

(19)



Europäisches Patentamt

European Patent Office

Office européen des brevets



(11)

EP 0 834 822 A2

(12)

EUROPEAN PATENT APPLICATION

(43) Date of publication:

08.04.1998 Bulletin 1998/15

(51) Int. Cl.⁶: G06F 17/30

(21) Application number: 97307009.7

(22) Date of filing: 10.09.1997

(84) Designated Contracting States:

AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC
NL PT SE

Designated Extension States:

AL LT LV RO SI

(30) Priority: 04.10.1996 US 726853

(71) Applicant:

Canon Information Systems, Inc.
Costa Mesa, CA 92626 (US)

(72) Inventors:

- Nehab, Smadar
Costa Mesa, California 92626 (US)
- Wickramaratne, Manjula G.
Costa Mesa, California 92626 (US)
- Klark, Paul L.
Costa Mesa, California 92626 (US)

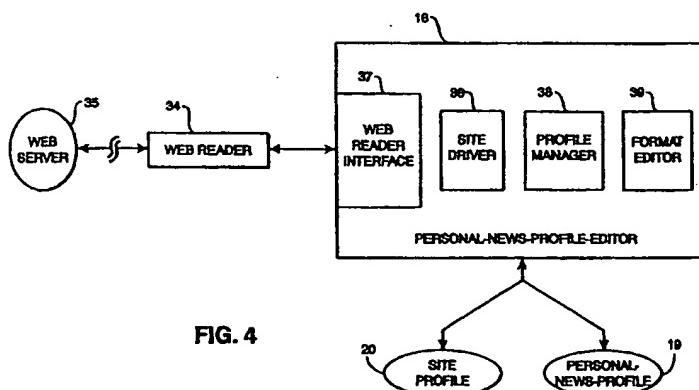
(74) Representative:

Beresford, Keith Denis Lewis et al
BERESFORD & Co.
2-5 Warwick Court
High Holborn
London WC1R 5DJ (GB)

(54) World wide web news retrieval system

(57) A World Wide Web site data retrieval system includes an input device for inputting data and commands to access the World Wide Web, and a memory for storing a Web site data retrieval driver which includes a Web reader, stored Web site address information, stored Web site commands, and stored format information. The memory also stores process steps to connect to a Web site and to issue commands within the connected Web site, and a connection to the World Wide Web. The system includes a processor for launching the Web site data retrieval driver in response to a command to access the World Wide Web. The Web site retrieval driver, upon being launched, (1) launches the

Web reader to connect to the World Wide Web via the connection, (2) retrieves the Web site address information and Web site commands, (3) instructs the Web reader to access the Web site based on the Web site address information and Web site commands, (4) downloads Web site data from the Web site based on the Web site commands, (5) stores the Web site data in a linear document, (6) repeats steps 1 through 5 until all addresses in the stored Web site address information have been accessed, and (7) formats the linear document into a personalized document based on the format information.



EP 0 834 822 A2

BEST AVAILABLE COPY

Description**BACKGROUND OF THE INVENTION****5 Field Of The Invention**

The invention relates to a data retrieval system which automatically traverses hypermedia documents on a computer network and automatically retrieves information from those documents based on a match between the structure of the documents and a personalized data retrieval structure. More particularly, the invention can retrieve articles from a news service, from a magazine service, or from a combination of both services which are located on the World Wide Web, a private computer network that supports hypermedia links, or any other hypermedia-linked computer system.

For example, there exists a Web site for retrieving news articles from the New York Times and a Web site for retrieving articles from People magazine. The retrieval system of the invention can traverse through such Web sites and select articles based on a personalized data retrieval structure. The personalized data retrieval structure may include commands to retrieve a full text of the front page only, headlines of the business section, headlines of the stock section and sports section, etc. In addition, the personalized data retrieval structure may include content-based rules to retrieve articles with certain keywords, to exclude articles with certain keywords, or to include articles based on a rule-based content analysis. The invention also provides a method for synthesizing all retrieved news articles and printing the synthesized news articles into a newspaper-type format in which each of the articles is arranged based on a user's pre-defined layout.

While the above example is in the context of the Web, hypermedia documents can reside on other types of networks besides the Web, such as an intranet. An intranet is a private computer network that is not connected to outside computer networks. For example, a company's own computer network could be an intranet with hypermedia documents on it. For brevity, the following discussion is made with respect to the World Wide Web. However, it should be understood that the invention applies equally well to any type of computer network that contains hypermedia documents, such as an intranet, different hypermedia-linked computer networks that reside on the Internet other than the Web, etc.

A hypermedia document on the Web can span multiple Web sites. Such documents can be newspapers, news articles, magazines, catalogs, manuals, memoranda, and the like. For brevity, the following discussion is made with respect to sources of news information. However, it should be understood that the invention applies equally well to any other type of hypermedia document.

Description Of The Related Art

The World Wide Web is an on-line source of hypermedia documents containing hypermedia text and images that act as links to other documents, Web sites, etc. As a result, documents on the Web are not organized sequentially. Rather, a user is automatically linked to other documents or Web sites to complete the viewing of a document by selecting a hypermedia link, such as a text link or an image link, within the document. Accordingly, an entire document cannot be viewed by scrolling through text.

One popular use of the Web is on-line publication and distribution of magazines and newspapers. Currently, many Web news services, such as the New York Times, allow the user to define keywords of interest and to receive news information, daily or hourly, that contains text matching the keywords. The news information can then be delivered to the user's computer via modem or E-mail. However, most Web news site newspapers, like the New York Times, include too much information, most of which has no interest to the user since the information is retrieved based only on a keyword match.

Other sources of news information are provided through information suppliers like "Individual Inc." Individual Inc. supplies users with a brief summary of the top twenty most relevant articles based on a user's predefined keywords. This subscription news service allows the user to specify five to ten areas of interest based on keywords, which are then prioritized by the user. The information service searches the Web for magazines and newspapers which contain any of the keywords. Based on the keyword searches, twenty of the most relevant articles are selected, compiled into a brief one-page summary, and transmitted to the user via facsimile for the user's review. However, in order to review an entire document rather than the summary, the user must log onto a specific Web site containing the document in order to retrieve and review the document.

There are yet other services which permit the user to personalize a newspaper to be displayed at the user's terminal by storing links to various news articles from various news sources on the Web. For example, CRAYON "Create Your Own Newspaper" permits a user to select specific sections from among links to over twenty-five different on-line newspapers, and to compose the selections into a personalized newspaper. Using CRAYON, it is possible to compose a personalized newspaper containing, for example, links to the international section of the New York Times, the business section of the Wall Street Journal, and the sports section of the Chicago Tribune. The HTML (hypertext markup lan-

guage) source file for this newspaper is then stored to mass media storage for later use.

While the forgoing news and information services provide convenient ways to keep updated on the news, they do not allow a user to access and view the news in the way that people naturally read a real-world newspaper. Namely, people naturally read a newspaper by scanning the pages of sections that they find interesting and then reading those articles that grab their attention. In other words, people use a structural approach to decide what pages to look at initially (e.g., the first page of the Business and World sections, and the comics page of the Arts section). They then scan the selected pages for articles.

In sum, conventional news and information services do not allow a user to access data from a hypermedia document on the basis of the structure of the document, and then to format that data in a manner that allows the user to scan and read the data in a natural fashion.

SUMMARY OF THE INVENTION

In accordance with one aspect the invention addresses the above deficiencies in the art by accessing at least one hypermedia document, retrieving data from the hypermedia document into an extracted data tree, with the data retrieved based on a structure of the hypermedia document, flattening the extracted data tree into a linear document, and formatting the linear document into a formatted document.

In another aspect, the invention creates a personal-news-profile for retrieving data from a hypermedia-linked computer network. The hypermedia-linked computer network is accessed, a learning mode is started, the hypermedia-linked computer network is traversed with commands, at least one rule is extracted from the commands, and the rule(s) is compiled into the personal-news-profile.

In yet another aspect, the invention creates a personalization profile for a Web site retrieval data retrieval system. Data and commands are input to access the World Wide Web and a connection is made to the World Wide Web. A Web reader is launched, and the Web reader accesses the Web via the connection. In response to user commands, a learning mode is entered into. Commands are sent to traverse the World Wide Web, and at least one rule is extracted from the commands. The rule(s) is compiled into a personalization profile, which is stored.

In yet another aspect, the invention retrieves articles from a hypermedia-linked computer network and formats the articles into a personalized newspaper. A stored personal-news-profile is retrieved. The personal-news-profile includes address data for a site on the hypermedia-linked computer network, command data for accessing data from the site, and newspaper layout commands. The site is accessed based on address data stored in the personal-news-profile, and articles at the site are downloaded based on command data stored in the personal-news-profile. The downloaded articles are flattened into a linear document, and the linear document is formatted into the personalized newspaper according to newspaper layout commands stored in the personal-news-profile.

In yet another aspect, the invention retrieves data from a World Wide Web site and formats the data into a personalized document. A Web site data retrieval driver which includes a Web reader, stored Web site address information, stored Web site commands, and stored format information is accessed. The invention (1) launches the Web reader to connect to the World Wide Web via a connection to the Web, (2) retrieves the Web site address information and Web site commands, (3) instructs the Web reader to access the Web site based on the Web site address information and Web site commands, (4) downloads Web site data from the Web site based on the Web site commands, wherein the data is downloaded with reference to a linked list so as to avoid hypermedia-links that form loops and so as to avoid repetitive downloading of data that has already been downloaded, (5) stores the Web site data in a linear document, (6) repeats steps 2 through 5 until all addresses in the stored Web site address information have been accessed, and (7) formats the linear document into the personalized document based on the format information.

In yet another aspect, the invention accesses and retrieves data at World Wide Web sites and formats the data into a personalized document. The invention connects to the World Wide Web, retrieves user defined Web site address information, user defined Web site commands, and user defined formatting commands, and activates a Web reader so as to access a Web site based on the user defined Web site address information. The Web reader is used to download data from the Web based on the user defined Web site commands, and the data is downloaded into an extracted data tree. The downloading continues until all addresses in the user defined Web site address information have been accessed. The extracted data tree is flattened into a linear document, and the flattened document is formatted into the personalized document based on the user defined formatting commands.

In yet another aspect, the invention retrieves news articles from on-line news services on the World Wide Web and formats the news articles into a personalized newspaper. The invention stores a personal-news-profile which comprises addresses data and command data for accessing data from a Web site and newspaper format commands, retrieves the stored personal-news-profile and accesses the data stored therein, activates a Web reader to contact a Web site based on address data stored in the personal-news-profile, downloads news articles at the contacted Web site based on command data stored in the personal-news-profile, stores the downloaded news articles, and formats the stored news articles into the personalized newspaper based on the newspaper format commands stored in the personal-news-profile.

In yet another aspect, the invention formats a hypermedia document into a personalized document. A location of the hypermedia document is specified, a type of the hypermedia document is specified, a scope of data to be retrieved from the hypermedia document is specified, wherein the scope is based on a structure of the hypermedia document, and a format is specified for formatting the data retrieved from the hypermedia document into the personalized document. The hypermedia document found at the specified location is accessed, data is retrieved from the hypermedia document in accordance with the specified hypermedia document type and in accordance with the specified scope, and the data is formatted into the personalized document in accordance with the specified format.

In yet another aspect, the invention is a system for processing a hypermedia document. The system accesses the hypermedia document, extracts addresses from the hypermedia document, and stores the addresses extracted from the hypermedia document in a container. The system activates a processing function to process data stored at the addresses stored in the container, downloads the data stored at the addresses stored in the container into a memory, and extracts predetermined data from downloaded data in accordance with predetermined configuration information. The predetermined data is then formatted in accordance with predefined formatting settings to generate a formatted document, and the formatted document is processed in accordance with the processing function.

In preferred embodiments, the system inputs the formatting settings and configuration information via a graphical user interface. The graphical user interface comprises plural processing icons, one of which activates the processing function. By virtue of the graphical user interface, a user can interactively set a document's format and change that format should a change be desired.

In particularly preferred embodiments, the graphical user interface is displayed in plural modes. The plural modes comprise (1) a fully-functional mode in which the graphical user interface displays formatting fields, processing options, menus and the processing icons, and (2) a minimizing mode in which the graphical user interface displays only the processing icons. Typically, the graphical user interface displayed in the minimizing mode is displayed during browsing the hypermedia document. By displaying the graphical user interface in plural modes, the present invention facilitates operation of the invention during browsing of the hypermedia document.

This summary has been provided so that the nature of the invention may be understood quickly. A more complete understanding of the invention can be obtained by reference to the following detailed description of the preferred embodiments thereof in connection with the attached drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 is a perspective view showing the outward appearance of the personal news retrieval system according to an embodiment of the invention.

Figure 2 is a block diagram of the personal news retrieval system shown in Figure 1.

Figure 3, comprised of Figures 3A, 3B, 3C and 3D, shows representational diagrams illustrating an example of the transformation of information from the Web (Figure 3A) to an extracted data tree (Figure 3B), then to a flattened document (Figure 3C), and finally to a formatted document (Figure 3D) according to an embodiment of the invention.

Figure 4 is a representational block diagram of the manner by which a personal-news-profile for retrieving news articles via the Web is created or edited according to an embodiment of the invention.

Figure 5, comprised of Figures 5A and 5B, shows flow diagrams describing how a personal-news-profile is created or edited.

Figure 6 is a representational block diagram of the manner by which news articles are retrieved from the Web and formatted with reference to a personal-new-profile according to an embodiment of the invention.

Figure 7 is a flow diagram describing how news articles are retrieved from the Web with reference to a personal-news-profile.

Figure 8 is a flow diagram showing how retrieved news articles are formatted with reference to a personal news profile and sent to a print device interface.

Figures 9A to 9E depict a graphical user interface used with the second embodiment of the present invention.

Figure 10 is a flow diagram describing the operation of the second embodiment of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Figure 1 is a view showing the outward appearance of a representative embodiment of the invention. Shown in Figure 1 is computing equipment 1, such as a Macintosh or an IBM PC or a PC-compatible computer, having a windowing environment, such as Microsoft Windows. Provided with computing equipment 1 is display screen 2, such as a color monitor or a monochromatic monitor, keyboard 3 for entering text data and user commands, and a pointing device such as mouse 4 for pointing and for manipulating objects displayed on display 2. Computing equipment 1 also includes a mass storage device such as disk drive 5. Image data can be input into computing equipment 1 from a variety of sources such as a network interface 11a or from external devices via facsimile/modem interface 6. Network interface

11a is used to connect computing equipment 1 to a local area network (LAN) or to a wide area network (WAN) such as the World Wide Web.

Figure 2 is a detailed block diagram showing the internal construction of computing equipment 1. As shown in Figure 2, computing equipment 1 includes central processing unit (CPU) 8 interfaced with computer bus 9. Also interfaced with computer bus 9 is printer interface 10, fax/modem interface 6, display interface 11, network interface 11a, keyboard interface 12, mouse interface 13, main memory 14, and disk drive 5.

Main memory 14 interfaces with computer bus 9 so as to provide random access memory storage for use by CPU 8 when executing an application such as personal-news-profile editor 16 or Web printer 17. More specifically, CPU 8 loads these software applications from disk drive 5 into main memory 14 and executes the software applications out of main memory 14. In accordance with user instructions, stored application programs are activated which permit processing and manipulation of data. Typically, the software applications stored on disk drive 5, such as personal-news-profile editor 16, Web printer 17, and HTML formatter 18, have been stored on disk drive 5 by downloading the software applications from a computer-readable medium such as a floppy disk or CD ROM, or by downloading the software applications from a computer bulletin board.

Disk drive 5 stores data files which can include text files and image files, in compressed or uncompressed format, and stores software application files such as those noted above. The software application files include Windows applications, DOS application, and personal news retrieval files 15. Personal news retrieval files 15 include personal-news-profile editor 16, Web printer 17, HTML formatter 18, personal-news-profile(s) 19, and site profile(s) 20. The detailed functions of personal news retrieval files 15 will be discussed below, after a brief overview of the operation of the personal news retrieval system.

Overview of Document Retrieval

Figure 3, comprised of Figures 3A to 3D, illustrates the operation of a representative embodiment of the invention.

Figure 3A is a graphical representation of a typical Web site 21 with news information contained therein. Within Web site 21 is homepage 22 with links to indices such as headings 23, which are in turn linked to articles 24. Some of articles 24 are linked to other articles. As article H 26 resides on another Web site, link 25 is a cross-site link. Link 25 illustrates how a single hypermedia document, represented by homepage 22, can traverse multiple Web sites.

In order to retrieve news from Web site 21, the invention first traverses Web site 21 to retrieve data according to user-defined rules. As will be discussed in more detail below, these rules can be based on the structure of Web site 21, or on the structure of Web site 21 and its contents. The data is retrieved into an extracted data tree, which preserves the organization of the data as shown in Figure 3B, but in which some links are excluded.

The organization of extracted data tree 27 has several features. First, extracted data tree 27 has root 28 which can have child nodes for one or more sites 29, which in turn can have index nodes 30 which correspond to indices/headings 23, articles nodes 31, and the like. Second, extracted data tree 27 is a true tree, with no loops (i.e., cyclic paths) therein. For example, Figure 3A shows a loop from homepage 22 to index node #1, to article C, and then back to homepage 22. This loop is removed when creating extracted data tree 27.

Second, the organization of extracted data tree 27 depends on how the Web sites are traversed, and not on the Web sites' actual layouts. Thus, article H 26 appears under index node #3 (under site #1), indicating that the news retrieval system accessed article H 26 from site #1 via cross-site link 25.

Finally, as noted earlier, certain articles have been excluded from extracted data tree 27 due to the structure of Web site 21 or possibly a content of indices/headings 23 and articles 24. For example, articles E and G have been excluded from extracted data tree 27.

According to this embodiment of the invention, extracted data tree 27 is flattened into linear document 32, as shown in Figure 3C, possibly with reference to more exclusion rules. Linear document 32 is simply a continuous document with information from extracted data tree 27 embedded therein.

Finally, linear document 32 is formatted according to user-specified (or default) formatting instructions into formatted document 33, shown as a stylized personal newspaper in Figure 3D. Formatted document 33 has various fonts and/or colors for site labels, indices/headings, articles, and the like. Furthermore, formatted document 33 is broken down into pages.

Note that in alternate embodiments of the news retrieval system, certain stages of the above transformation from Web site 21 to formatted document 33 can be skipped. For example, data from Web site 21 can be retrieved directly into flattened document 32, as long as a record of the organization of the data is maintained (possibly in a separate linked list) so as to avoid downloading the same article twice and so as to avoid loops in the organization of Web site 21. Alternatively, extracted data tree 27 can be directly formatted into formatted document 33. In any case, the basic operation of the invention remains the same: the news retrieval system traverses a hypermedia document on the Web, extracts data according to user-defined information, and formats the data into a personalized newspaper.

As mentioned in the above discussion, various user-defined rules and other information (such as formatting infor-

mation) are involved in the news retrieval process. That user defined information is stored in personal-news-profile(s) 19, the definition of which is described next.

Defining a Personal-News-Profile

5 Figures 4 and 5 illustrate the process by which personal-news-profile 19 is defined. To create personal-news-profile 19, personal-news-profile editor 16 communicates with personal-news-profile 19, site profile 20, and Web reader 34.

Personal-news-profile 19 contains information as to what sites to access for creating a personalized newspaper, what sections to retrieve from those sites, rules to be used to determine what data to extract from the sections and the 10 articles therein, rules to determine how to exclude links, and newspaper format information. A sample personal-news-profile is shown in Appendix 1.

Site profile 20 includes general site information that is not specific to a particular user. For example, site profile 20 could contain information such as full site addresses, sections within a site, non-user specific passwords, etc. Sample 15 site profiles are shown in Appendix 1. Because general site information is stored in site profile 20, personal-news-profile 19 can refer to the general site information with reference to site profile 20, saving space in the personal-news-profile. For example, as shown in Appendix 1, personal-news-profile 19 can refer to a site number 1. Site profile 20 indicates that site number 1 is the "San Jose Mercury News," with a homepage at "http://www.sjmercury.com/". This construction also centralizes general site information. Thus, if a site address changes, only site profile 20 needs to be changed to update all personal-news-profiles 19 on the system.

20 Web reader 34 is an application program or program module that communicates with the Web via Web server 35. In response to commands from personal-news-profile editor 16, Web reader 34 will access the Web, traverse hypermedia 25 documents on the Web, retrieve data from the documents, and return the retrieved data to personal-news-profile editor 16.

As shown in Figure 4, personal-news-profile editor 16 includes four modules: site driver 36, Web reader interface 25 37, profile manager 38, and format editor 39.

Web reader interface 37 interfaces personal-news-profile editor 16 to Web reader 34. Site driver 36 interacts with 30 Web reader 34 via Web reader interface 37 to provide an abstract interface to each individual Web site. More specifically, site driver 36 instructs Web reader 34 to access various Web sites and to retrieve data from those sites. Thereafter, site driver 36 receives that data and builds site profile 20 therefrom. The date can also be used to update an existing site profile.

In building site profile 20, site driver 36 translates the structure of each accessed Web site to a uniform structure 35 defined in site profile 20, and stores data retrieved therefrom in site profile 20. By translating different Web sites, some of which may have different structures, into a single uniform structure and storing data therefrom in that structure in site profile 20, the present invention facilitates access to information from different Web sites, and thus reduces overall processing time.

Profile manager 38 maintains document templates that specify how to format a personalized newspaper. Predefined 40 document templates exist. In addition, format editor 39 allows a user to specify personalized templates for formatting a newspaper, either by editing existing templates or by creating new ones. In any case, each document template specifies page layout information, font information, style information, colors, etc. for the titles, indices/headings, sub-headings, text and the like for a personalized newspaper.

Sample code for personal-news-profile editor 16, site driver 36, and profile manager 38 is included in Appendix 3A.

Figures 5A and 5B are flow diagrams describing the operation of personal-news-profile editor 16 in more detail. Figure 5A shows the operation of personal-news-profile editor 16 in defining the parts of personal-news-profile 19 relating to accessing Web sites and retrieving data from those sites.

45 In step S500 of Figure 5A, personal-news-profile editor 16 is launched by a user. In step S501, the editor launches Web reader 34. The user's personal I.D. is then retrieved in step S502. If a personal-news-profile already exists for that I.D., step S503 directs flow to step S504, where the user is given the option of skipping to the format editor. Otherwise, personal-news-profile editor 16 enters a "learning mode" in step S505. Once in the learning mode, personal-news-profile editor 16 proceeds to step S506, where it accepts a Web command (i.e., a command to traverse a hypermedia link) 50 from the user and forwards the Web command to the Web reader by means of site driver 36. Site driver 36 maintains a hierarchical log of Web sites visited by Web reader 34. In step S507, personal-news-profile editor 16 creates an extraction rule from the Web command. This rule will allow the news retrieval system to later duplicate the user's selection criteria in browsing (clicking on hyperlinks within) a Web site.

The rule specifies, at the least, structural criteria for duplicating the traversal of the Web site. For example, if a user 55 accesses all articles under a particular index/heading, the rule will specify that all articles under that index/heading should be retrieved.

In one embodiment of the invention, the rule can also include content-based criteria (i.e., keyword-based criteria) accepted from the user. These content-based rules can, for example: (1) require certain words to be in an article, (2)

exclude articles with certain words, (3) require certain boolean combinations of words, (4) rank articles that are selected based on structural criteria, with the ranking based on keywords, and then require the selection of the articles with the highest ranking(s), or (5) exclude certain types of articles such as advertisements.

- Examples of the syntax for the structural and content-based exclusion rules are shown in Appendix 2. Several different types of rules are shown. Some simply limit the traversal of a Web site to a certain number of links. Others are date and keyword based exclusion rules. One particularly flexible rule indicates that articles should be ranked based on a keyword analysis and the top scoring articles should be chosen. Other rules include "flattening" rules. These rules control the flattening of the extracted data tree, as will be explained in more detail below.

- At the least, the rule includes structural information about the user's selection (i.e., first page, first document, all links, etc.), necessary password information, browser commands, and the like. The rule can also include a pointer or a reference to site profile 20 and the appropriate information therein. General (non-user specific) information is used by site driver 36 to maintain site profile 20. In this manner, address information and passwords common to multiple users can be maintained in site profile 20, as discussed above. For example, site driver 36 will store commands or hyperlinks to other documents in a Web page in the rule, but will not store a Web site's full address in the rule. That address information is stored in site profile 20.

- In step S508, rule data defining the rule created from a Web command(s) is stored in an extracted data tree such as extracted data tree 27 in Figure 3B. This data tree is a linked list that reflects the organization of the data retrieved from the Web. In step S509, flow returns to step S506 for the next Web command unless the user is done (i.e., the user signs-off the Web site), in which case flow proceeds to step S510.

- At this point, the creation of the personal-news-profile has proceeded much like the creation of a macro common to word processing programs, except that site profile 20 has been used to minimize storage requirements and to centralize general site information. In order to minimize storage requirements further and in order to make the news retrieval system more flexible and efficient, the extracted rules are now compiled to remove redundant links, multiple visits to the same site, and the like. This occurs in step S510, and the resulting compiled rules become the first part of personal-news-profile 19.

- Alternatively, personal-news-profile editor 16 may be invoked as a graphical user interface which allows a user to edit a previously stored personal-news-profile or to specify document composition preferences, for example, by specifying news sites, headline articles only, keywords, etc. In either case, the result is personal-news-profile 19, which comprises a listing of Web site pointers as well as extracted rules for traversing through a Web site or sites.

- For a better understanding of the above, sample personal-news-profiles and sample site profiles are provided in Appendix 1 as noted above.

- Next, operation proceeds to give the user an option to modify a custom newspaper template, as shown in Figure 5B. In step S511, it is determined if a newspaper template has been defined and stored in personal-news-profile 19. If a newspaper template has been defined, step S512 gives the user the option to edit the template or to proceed to step S520. If the user chooses to edit the template or if no newspaper template has been defined, flow proceeds to step S513.

- Step S513 gives the user the option of creating a custom template or using a predefined template. If the user wants to use a predefined template, step S514 gets the specified predefined template, which is added to the personal-news-profile in step S519. Otherwise, flow proceeds to step S515, where format editor 39 is invoked.

- Format editor 39 has a graphical user interface that provides the user with a number of formatting options. In step S516, format editor 39 allows the user to define which newspaper sections are to be printed in the newspaper, which Web site's news article are to be placed in each section, and/or how each page is to be laid out. In this regard, the user can specify which Web site's news articles are to be used as a front page, which Web site's news articles are to be used as a business page, which Web site's news articles are to be used as a sports page, etc. In addition, in step S516, the user can define where each index/heading should be listed, as well as what sub-headings should go on each page.

- In step S517, format editor 39 allows the user to define the font styles for indices/headings, sub-headings, bylines and actual text of news articles. In step S518, format editor 39 prompts the user to define index/heading colors, title colors, etc. In this regard, layout editor 39 is capable of determining the types of fonts and colors available to the user based on the system's printer capabilities.

- Once all of the information is gathered for the custom template, the format editor adds the information to personal-news-profile 19 in step S519. Alternatively, profile manager 38 may also store the custom format as a template in a common area for use by other users. In this case, only a pointer or reference to the custom template is stored in personal-news-profile 19.

- In step S520, personal-news-profile editor 16 prompts the user to set an automatic newspaper delivery time and method (i.e., print or store on disk drive 5 for later printing). These settings are added to personal-news-profile 19. More specifically, in the case that a user's computer is continuously supplied with power, the Web news retrieval system can be launched automatically at a designated time. The system will retrieve articles from the Web sites which are listed in personal-news-profile 19. Upon retrieving the news articles, the articles will be formatted based on the newspaper tem-

plate in personal-news-profile 19. The formatted personalized newspaper can then be either printed or stored for later viewing. In the case that a time is not set for newspaper delivery, the user can execute the Web news retrieval system program at any time.

Once personal-news-profile 19 has been created, the Web news retrieval system, upon being launched, can traverse Web news sites and build a personalized newspaper by automatically retrieving various news articles from the Web news sites and print the news articles based on the newspaper template indicated in personal-news-profile 19. A description of how the Web news retrieval system of the invention performs this function is described next.

Retrieving a Document Using a Personal-News-Profile

Figure 6 is a representational block diagram of the manner by which the invention retrieves articles from the Web according to personal-news-profile 19. (Figure 6 also shows the manner by which the retrieved articles are flattened into a linear document and formatted. These functions are discussed in greater detail in the next section of this application.)

As shown in Figure 6, Web printer 17 is responsible for retrieving news articles. Web printer 17 is an end-user application that communicates with personal-news-profile(s) 19, site profile 20, Web reader 34, and output interface 40 in order to perform this function.

Web printer 17 looks at personal-news-profile 19 to determine which Web sites to access and which data to retrieve from those sites. Web printer 17 also looks at site profile 20 for general site information. According to the information in personal-news-profile 19 and site profile 20, Web printer 17 instructs Web reader 34 to connect to the Web via Web server 35 in order to access various Web sites and to retrieve data from those sites. Web reader 34 sends the retrieved data to Web printer 17, and Web printer 17 uses the data to build an extracted data tree. As will be discussed in greater detail in the next section of the application, Web printer 17 then flattens the extracted data tree into a linear document and formats the linear document for output via output interface 40.

As shown in Figure 6, Web printer 17 includes four program modules: Web reader interface 50, site driver 51, tree manager 41, and formatter 42.

Web reader interface 50, like Web reader interface 37 described above, interfaces Web printer 17 to Web reader 34.

Site driver 51 accesses site profile 20 and personal-news-profile 19 and provides data stored therein to Web reader 34. As noted above, Web reader 34 uses that data to access various Web sites and to extract data therefrom. As noted above, this retrieved data is used by Web printer 37 to build an extracted data tree.

Tree manager 41 manages the extracted data tree. In this regard, tree manager 41 keeps track of the organization of the retrieved data in the extracted data tree. This allows Web printer 17 to avoid accessing the same article twice, to avoid unnecessarily re-visiting a Web site, and to avoid getting caught in a cycle (loop) in the organization of a hypermedia document on the Web. Alternatively, tree manager 41 could store the data in blocks (as opposed to directly in a data tree) with reference to a linked list that provides the same functionality as the extracted data tree. Sample code for tree manager 41 is included in Appendix 3B.

Formatter 42 is responsible for flattening the extracted data tree into a linear document and formatting the linear document into a personalized newspaper. Formatter 42 performs these functions in accordance with the print criteria and format information (i.e., newspaper template) indicated in personal-news-profile 19. Sample code for formatter 42 is included in Appendix 3B.

In more detail, Figure 7 is a flow diagram describing how Web printer 17 uses Web reader 34 to traverse the Web according to personal-news-profile 19 and to retrieve articles from the Web according to the profile, excluding unwanted data.

The Web printer starts in step S700. In step S701, Web printer 17 retrieves either a user designated personal-news-profile or a default personal-news-profile stored in disk drive 5 using site driver 51. In this regard, because computer equipment 1 may be used by more than one user, there may be one or more personal-news-profiles stored on the equipment, one of which will be designated as the default. Upon retrieving the designated personal-news-profile, in step S702 Web printer 17 determines whether any news data has been previously stored to disk drive 5 (for example, by a previous automatic news delivery) or if news articles should be retrieved using personal-news-profile 19.

In the case that news data does exist on disk drive 5, in step S703 the stored news data is retrieved and flow proceeds to step S801 of Figure 8, discussed in more detail in the next section. On the other hand, if no stored news data exists, Web printer 17 invokes Web reader 34 in step S704. Note that this is the same Web reader 34 as discussed above with respect to defining a personal-news-profile.

Upon being invoked, Web reader 34 connects to Web server 35 in step S705, which provides a connection to a network, such as the World Wide Web. Web printer 17 then provides Web reader 34 with an address for the first Web site to be visited based on information retrieved from personal-news-profile 19. Once connected to the desired Web site in step S706, Web printer 17 provides Web reader 34 with commands/links for traversing the Web to the next Web page containing information that personal-news-profile 19 indicates should be retrieved. Web reader 34 traverses the Web

according to this information in step S707.

In step S708, Web reader 34 retrieves the desired information and sends it to Web printer 17 according to the rules in personal-news-profile 19. Thus, data exclusion occurs in this step. The rules in personal-news-profile 19 specify structural and content-based criteria for excluding data from the personalized newspaper. The structural rules limit the retrieved information on the basis of the structure of the Web site accessed by Web reader 34. The content-based rules limit the retrieved information on the basis of its content. As mentioned above with respect to creating a personal-news-profile, examples of the syntax of the retrieval rules in personal-news-profile 19 are included in Appendix 2.

In addition to rule-based exclusion, media-type exclusion occurs in step S708, wherein data of a media type that can not be printed is excluded from the extracted data tree. For example, movie and sound data can be excluded.

Web printer 17 stores the retrieved data in disk drive 5 (or in main memory 14) in the extracted data tree managed by tree manager 41. Alternatively, the data could be stored in blocks with reference to a linked list, as discussed earlier. In step S709, Web printer 17 returns to step S707 to complete retrieving all information from Web pages at the Web site. In step S710, upon completing a traversal of one Web site, Web printer 17 uses tree manager 41 to compare the sites remaining in personal-news-profile 19 with the site organization information in the extracted data tree to determine if more sites need to be visited. In the case that more Web sites need to be visited, step S710 returns flow to step S706 and news articles are retrieved in the same manner as discussed above. On the other hand, if all of the Web sites listed in personal-news-profile 19 have been visited and all of the articles retrieved, flow proceeds to step S801 in Figure 8.

Flattening and Formatting the Retrieved Data

Figure 8 is a flow diagram showing how the extracted data tree is flattened and formatted. The configuration of the invention is the same as when retrieving data from the Web (shown in Figure 6). In fact, the flattening and formatting processes can occur, at least to a limited extent, concurrently with the data retrieval process.

In step S801 of Figure 8, the extracted data tree is flattened. This simply means that the organization of the data is converted from an extracted data tree to a linear document. This step provides the opportunity for excluding more data from the personalized newspaper, for example by only including nodes of the data tree into the flattened document. This exclusion process is controlled by the flattening rules in personal-news-profile 19.

After the data is flattened into a linear document, the data is formatted in step S802 according to the template indicated in personal-news-profile 19. The definition of this template, which is either a pre-defined template or a custom template, was discussed earlier. Finally, in step S803, the formatted and fully personalized newspaper is sent to output interface 40. This interface could be printer interface 10 to printer 7, display interface 11 to display 2, or even modem/fax interface 6.

Second Embodiment: The HTML Formatter

The second embodiment of the invention is a system for processing a hypermedia document. The system accesses the hypermedia document, extracts addresses from the hypermedia document, and stores the addresses extracted from the hypermedia document in a container. The system activates a processing function to process data stored at the addresses stored in the container, downloads the data stored at the addresses stored in the container into a memory, and extracts predetermined data from downloaded data in accordance with predetermined configuration information. The predetermined data is then formatted in accordance with predefined formatting settings to generate a formatted document, and the formatted document is processed in accordance with the processing function.

The second embodiment of the invention is depicted as HTML formatter 18, noted in Figure 2. An example of HTML formatter 18 is WebFormatter, manufactured by Canon Information Systems, Inc. The second embodiment will be described with respect to WebFormatter. It should be noted, however, that HTML formatter 18 is not limited to the WebFormatter embodiment, and that various alternative embodiments within the spirit and scope of the following description are possible.

WebFormatter is stand-alone utility software that can be used in conjunction with different Web browsers, such as Netscape, Mosaic and Internet Explorer. In short, WebFormatter extracts data from a Web page, strips out extraneous data from the extracted data, and reformats the data into a formatted document. The formatted document can then be printed, stored in an RTF (Rich Text Format) file, or edited in any RTF compatible editor, such as MS Word, WordPerfect, Wordpad, etc.

WebFormatter can be activated from a windowing environment, such as Microsoft Windows®. From such a windowing environment, WebFormatter can be activated by double-clicking on a WebFormatter icon (not shown) in a start-up window, selecting WebFormatter from the Windows start menu, dragging a URL (uniform resource locator) icon (not shown) from a Web browser and dropping it into the WebFormatter icon, or by automatically invoking WebFormatter when the Web browser is started.

Unlike the first embodiment of the invention described above, WebFormatter does not use a predefined personal-

news-profile to specify criteria for creating a particular type of document from one or more Web pages. Rather, WebFormatter relies upon user-specified criteria to create a particular type of document, such as a newspaper or the like, from one or more Web pages. These criteria are input interactively by a user via a graphical user interface.

As described in more detail below, WebFormatter operates in two modes - a minimized mode and a fully-functional mode. In the minimized mode, WebFormatter's graphical user interface is essentially a floating print button, which is displayed concurrently with displayed Web pages. By virtue of this feature, as a user explores the Web, the user can process, format, and print out Web pages by merely clicking on the floating print button.

In its fully-functional mode, WebFormatter's graphical user interface provides spaces for a user to enter a URL address of a Web page to be processed, enter a personal title for the document, select a format for the document, preview a formatted first page of the document, and either print the document, save the document as an RTF file, or view/edit the document using an RTF editor. The graphical user interface for the fully-functional mode will be described first, since it is from that interface that the user can enter the minimized mode.

Figure 9A shows graphical user interface 43 for WebFormatter's fully-functional mode. Graphical user interface 43 is displayed on display 2 upon first activation of WebFormatter. As with any interactive windowing software application, a user interacts with graphical user interface 43 by means of mouse 4 (by pointing and clicking) and keyboard 3.

As shown in Figure 9A, graphical user interface 43 includes fields 44 and 46 to 49, through which a user can specify the URL address of a document to be formatted and the format of that document. Beginning with URL field 44, a user enters the URL address (e.g., http://www.cis.canon.com/tis/tis_home.htm) of a Web page to be processed by WebFormatter. There are several different ways for the user to enter the URL address. The user can (1) type the address directly into URL field 44, (2) copy the URL address in the Web browser and paste the URL address into URL field 44, (3) drag the URL address from the Web browser onto graphical user interface 43 or onto the WebFormatter icon, or (4) click on Current URL button 54.

With regard to Current URL button 54, if a user clicks on Current URL button 54, WebFormatter locates the active Web browser and queries the Web browser for the address of the current Web page. Thereafter, the Web browser provides the address of the current Web page to WebFormatter, which places the address in URL address field 44. If URL button 54 is activated and no Web browser is currently running, WebFormatter displays dialog box 56, shown in Figure 9A.

As shown, dialog box 56 includes Cancel button 57 and Launch Browser button 59. Cancel button 57 cancels a user's request to input a URL address into URL address field 44 via Current URL button 54. Launch Browser button 59, on the other hand, launches a Web browser specified in WebFormatter. As noted below, WebFormatter is configured beforehand with predefined information including a Web browser to be used with WebFormatter. Configuration of WebFormatter will be described in more detail below.

In alternative embodiments of WebFormatter, a filename can also be entered into URL address field 44. For example, in these alternative embodiments, if a user wishes to format a hyper-linked manual into a book-like format, the user enters the filename into URL address field 44. Thereafter, WebFormatter proceeds through the file in the same manner as through specified Web pages in order to reformat the hyper-linked manual as desired.

Returning to graphical user interface 43, title field 46 enables a user to enter a personalized title for a formatted document. The title may be typed directly or pasted into title field 46.

Formatting fields 47 to 49 define the format of a document to be output by WebFormatter. Options for the different formatting fields can be accessed by clicking on a scroll bar, such as scroll bar 55, of a respective formatting field. Each of these fields is described in detail below.

Styles field 47 provides four options for formatting an output document. These styles relate to characteristics of an output document such as size of headers, margins, etc. The style options include Contemporary, Formal, Fun and Professional. The invention, of course, is not limited to these four style options, and other styles can be added as desired.

Columns field 48 defines the number of columns in a formatted output document. Two columns options are available - Single and Multiple; however, the invention is not limited to these two options. The Single option, as might be expected, formats the document into a single column. The Multiple option, on the other hand, formats the document into a predetermined number of columns. In preferred embodiments of the invention, the multiple option is set to two columns; however, any number can be set.

Spacing field 49 defines the spacing between lines in a formatted output document. Three options are provided in WebFormatter, but other options can be added as desired. These three options are Condensed, Normal and Easy To Read, with Condensed being the least amount of spacing between lines and Easy To Read being the most amount of spacing between lines.

Graphical user interface 43 is also provided with Preview button 60. By clicking on preview button 60, a user can preview a first page of a formatted document in viewing area 61. An example of a previewed formatted document is shown in Figure 9A.

As shown in Figures 9A and 9B, WebFormatter also includes Options button 61. Options button 61 provides a user with additional formatting options which are used by WebFormatter to create a formatted document. A user can activate

Options button 61 by clicking thereon. This causes Options dialog box 62, shown in Figure 9B, to appear on display 2.

As shown in Figure 9B, options dialog box 62 includes General options 64, Container options 66 and Strip Meta Info options 67. General options 64 includes "Text only" listbox 72, "Index of links in the page" listbox 73, and "No floating pictures" listbox 74. These options are indicated as being selected by a check mark or the like in a respective listbox.

- 5 As will become clear from their descriptions, more than one of the options in General options 64 can be selected at the same time.

"Text only" listbox 72 instructs WebFormatter to strip all graphics in a Web page and print only text therein. "Index of links in the page" listbox 73 instructs WebFormatter to add a list of all URLs present in a Web page or pages to the end of a formatted document. Preferably, the list of URLs is printed as superscript, and anchor positions of the URLs in the list are marked in bold. "No floating pictures" listbox 74 instructs WebFormatter to print all images in the document in a particular area of the formatted document. In some cases, therefore, when this option is selected; WebFormatter shrinks images, as needed, so that images fit into a particular area.

- 10 Strip Meta Info options 67 provides engineering options which facilitate stripping of unnecessary information from a Web page being processed by WebFormatter. The options include (1) "None", which instructs WebFormatter to strip nothing from the Web page, (2) "Till the first horizontal rule", which instructs WebFormatter to strip all links and images until and up to predefined first and second horizontal formatting rules (e.g., up until a horizontal line across a page), and (3) "Till the first text", which instructs WebFormatter to strip all links and images up to first and last occurrences of text in the Web page. Only one of Strip Meta Info options 67 can be selected at a time. Selection thereof is indicated by a dot in a bullet located next to an option, as shown in Figure 9B.

- 20 Container options 66 provides options for processing documents, addresses for which are stored in container 76 shown in Figure 9B. Prior to describing Container options 66, a description of container 76 will be provided.

- 25 As noted, container 76 stores URL addresses of selected documents. Document addresses which are input to field 44 are added to container 76. The order in which URLs are input into container 76 denotes the order in which data in the URLs is processed by WebFormatter. As shown in Figure 9B, once container 76 becomes full, its icon changes to that shown by reference numeral 77.

When a user clicks on the icon for container 76, menu 77 is displayed. Menu 77 provides five options; i.e., Open 79, Empty 80, Print 81, Edit 82 and Save 84. These options are highlighted when activated, and are described in detail below.

- 30 Open 79, when activated, displays Container Contents screen 87 shown in Figure 9B. Container Contents screen 87 shows the URL addresses stored in container 76. Container contents screen 87 provides four buttons; i.e., Add current URL button 88 which adds the current URL to container 76, Delete button 89 which permits a user to highlight and delete a URL in container 76, Empty button 90 which permits a user to empty container 76, and Done button 91 which permits a user to close Container Contents screen 87. It is noted that a user can also empty the contents of container 76 by clicking on Empty 80 of menu 77.

- 35 In addition, the user can rearrange the order of URLs stored in container 76 by dragging and dropping different URLs at different locations therein. As noted above, since the URLs are processed in the order that they appear in container 76, this feature permits a user to rearrange the processing order of the URLs in container 76 interactively.

- Print 81, Edit 82 and Save 84, when activated, cause WebFormatter to download all data at Web pages defined by the URLs stored in container 76, format them as specified by the user, create RTF file(s) storing the formatted Web pages, and do the selected action, i.e., save, edit or print the RTF file(s). This process is described in greater detail below.

- 40 Referring back to Options dialog box 62, Container options 66 include "Print table of contents" listbox 92 and "Empty after processing" listbox 94. As shown, a check mark appears in a listbox to indicate that the listbox has been selected. In this regard, more than one listbox can be selected at a time. "Print table of contents" listbox 92, when selected, instructs WebFormatter to print titles of all URLs in container 76 as a table of contents in a formatted output document. "Empty after processing" listbox 94, when activated, instructs WebFormatter automatically to empty container 76 after printing, editing or saving a document, without waiting for a user to do so.

- 45 Also shown as part of Container options 62 are Select RTF Editor button 69, Cancel button 70 and OK button 71. By clicking on Select RTF Editor button 69, a user can select an RTF file editor, examples of which are noted above. This can be done, for example, by displaying another dialog box listing predefined RTF editors (not shown) and selecting one of the predefined RTF editors. Cancel button 70 cancels Container options 62 and OK button 71 confirms selected options in Container options 62 and then closes its dialog box.

- 50 As shown in Figure 9B, graphical user interface 43 also includes print icon 96, edit icon 97, save icon 99, help button 100, done button 101 and minimizing icon 102. A user may select any of these features by clicking thereon using a mouse.

Print icon 96 opens a print dialog box (not shown), which allows a user to print any number of copies of Web pages formatted by WebFormatter. Edit icon 97 opens an RTF file storing formatted Web page(s) for editing by a predetermined RTF editor. Save icon 99 opens a save dialog box (not shown), which allows the user to name and save a for-

matted Web page as an RTF file. Help button 100 provides help messages for operating WebFormatter, and Done button 101 exits from WebFormatter. Minimizing icon 102 activates the minimizing mode of Webformatter which was mentioned above and which is described in greater detail below.

Figure 9C shows menus provided by WebFormatter during its operation. These menus include file menu 103, edit menu 104 and window menu 106. File menu 103 provides "Save", "Edit" and "Print" options, the functions of which are identical to those of Save icon 99, Edit icon 97 and Print icon 96, respectively. An "Exit" option is also provided to exit from File menu 103. Finally, File menu 103 provides "Open HTML file" option 107. This option provides a user with the capability to open a local HTML file; i.e., a hypermedia file resident on the user's computer such as a file saved from NetScape, or URL files created by dragging and dropping a URL onto the windows desktop. "Open HTML file" option 107 also provides hooks needed to open files created by other Web-file-processing products so that those files can be formatted as RTF files and printed, saved and/or edited using WebFormatter.

Edit menu 104 provides "Paste URL" option 109. "Paste URL" option 109 pastes the contents of a paste buffer, such as a URL address copied from a Web page, into URL field 44, as described above.

Window menu 106 provides a "Help Topics" option which provides a user with information regarding the use, maintenance and background of WebFormatter, and an "About WebFormatter" option which provides a user with a dialog box (not shown) containing WebFormatter's version number and copyright notice(s). Window menu 106 also includes "Preferences" option 110. "Preferences" option 110 opens preferences dialog box 112, shown in Figure 9D.

Preferences dialog box 112 is used to configure and re-configure WebFormatter. As shown in Figure 9D, preferences dialog box 112 includes Minimize view options 113, General options 114 and WWW Browser to use options 115. Minimize view options 113 can be set to configure WebFormatter's graphical user interface in the minimized mode. Two sets of options are provided. The first set include "Print", "Edit" and "Save". These options correspond to print icon 96, edit icon 97 and save icon 99, shown in Figure 9B. When a check mark appears in a listbox next to one of these options, the icon for that option is displayed in the minimized mode, e.g., the print icon, the edit icon and/or the save icon. More than one option can be selected at once. In this regard, Figure 9E shows graphical user interface 116, which is a representative example of a graphical user interface for WebFormatter when WebFormatter is in the minimized mode.

Referring back to Figure 9D, Minimize view options 113 also include "Row" and "Stack" options. These options can be set to display WebFormatter's graphical user interface in the minimized mode horizontally by selecting "Row" or vertically by selecting "Stack". Only one of these options can be selected at a time. As an example of the foregoing, graphical user interface 116 corresponds to a row of icons.

WWW Browser to use options 115 determine which World Wide Web browser is to be used with WebFormatter. As shown, preferably NetScape, Internet Explorer and Mosaic are provided as browser options; however, other browser options can also be provided. As might be expected, only one of these options can be selected at a time. The default browser option is NetScape Navigator.

General options 114 include "Auto-start with browser" option 117, "Open in minimized view" option 118, "Warn before printing more than pages" option 119, and "Warn before saving more than ____ MBs" option 120. "Auto-start with browser" option 117 sets WebFormatter to be invoked automatically when a Web browser is activated. If this option is not selected (which is the default), WebFormatter is opened by double clicking on a WebFormatter icon in the windowing environment, selecting WebFormatter from the Windows start menu, or dragging and dropping a URL from the Web browser into the WebFormatter icon, as described in more detail above. "Open in minimized view" option 118, when selected, opens WebFormatter in minimized mode. The default, however, is the fully-functional mode. "Warn before printing more than ____ pages" option 119, and "Warn before saving more than ____ MBs" option 120 allow a user to control the number of pages saved of a formatted document and the amount of memory space used by those pages, respectively. The default for both of these options is for no warning to be given. As is evident, more than one of the general options can be selected at the same time.

Preferences dialog box 112 also includes cancel button 121 which cancels a user's selected preferences and OK button 122 which confirms a user's selected preferences.

As explained above, WebFormatter can be configured to enter directly into the minimized mode via Preferences dialog box 112, or a user can enter the minimized mode via minimizing icon 102 shown in Figure 9B. As also noted above, Figure 9E shows an example of graphical user interface 116 for WebFormatter in the minimized mode. Graphical user interface 116 is displayed as a floating interface while a user is exploring the Web. Thus, as a user views a Web page, the user also views graphical user interface 116. By clicking on an appropriate icon on graphical user interface 116 (which, in Figure 9E, includes icons identical in both structure and function to those shown in graphical user interface 43), the user can capture the current Web page, process and format the Web page into an RTF file, and save, edit and/or print the RTF file. Alternatively, the user can drag a URL from the Web browser and drop it into one of the icons.

A user can reconfigure WebFormatter in the minimizing mode by double clicking a right mouse button. This action causes a preferences dialog box to appear on display 2 which is identical to preferences dialog box 112. Thereafter, the user can alter the configuration of WebFormatter as desired. Should a user wish to enter the fully-functional mode from the minimizing mode, the user need merely click on maximizing icon 117 shown in Figure 9E.

Figure 10 is a flow diagram describing the operation of WebFormatter. WebFormatter is activated in step S1000. As described above, this can be done by double-clicking on a WebFormatter icon in a windowing environment. Depending upon how WebFormatter has been configured, i.e., in the fully-functional mode or the minimizing mode, either a graphical user interface similar to that of graphical user interface 43 or one similar to that of graphical user interface 116 is displayed in step S1000. For the sake of completeness, the following assumes that a graphical user interface similar to that of graphical user interface 43 is displayed in step S1000, since the default mode of WebFormatter is the fully-functional mode.

Next, in step S1001, WebFormatter is configured, as described above via preferences dialog box 112 and options dialog box 62. This step is not necessary unless a user wishes to change WebFormatter's previously set configuration. In step S1002, document format data is input in fields 44 and 46 to 49 described above. More specifically, the user inputs a URL (or filename in alternative embodiments) into URL field 44. As described below, WebFormatter uses this information to process Web pages stored at the URL to create an RTF file based on the configuration of WebFormatter and the data input in fields 46 to 49.

In step S1003, a Web reader similar to that of Web reader 34 described above is executed. The Web reader connects to a network, such as the World Wide Web, in step S1004.

Next in step S1005, it is determined whether a URL or a filename has been entered. As described above, in preferred embodiments of WebFormatter, only a URL may be entered. However, since alternative embodiments of WebFormatter may permit entry of a filename, a description of processing a file other than one at a URL address will be provided.

If a URL has been entered in field 44, processing proceeds to step S1006. In step S1006, the Web reader accesses the hypermedia document (e.g., a homepage) specified by the URL address. In step S1007, WebFormatter instructs the Web reader to traverse the hypermedia document. Thereafter, WebFormatter selects URL address(es) from the Web and stores the addresses in container 76. Once all desired addresses have been selected and a processing function, such as print, has been activated, WebFormatter downloads data stored at the addresses in container 76 into memory 5. WebFormatter then extracts predetermined data from the downloaded data based on the configuration information set in Optional dialog box 62, and stores the extracted data in memory 5. Thus, for example, if "Text Only" option 72 in Options Window 62 is on, only text is extracted from the downloaded data. Processing then proceeds to step S1011.

On the other hand, if, in step S1005, a filename for an HTML source file is entered, WebFormatter instructs the Web reader to access a first site in the file. In steps S1008 and S1009, the site is traversed and data is extracted and stored in the same manner as in step S1007, described above. Then, in step S1010, WebFormatter determines if more sites are listed in the HTML source file. If more sites are listed in the file, flow returns to step S1008, and the next site is accessed. If no more sites are present, processing proceeds to step S1011.

In step S1011, WebFormatter processes the extracted data in accordance with the previously set format information. For example, if Columns field 48 is set to multiple, the extracted data will be formatted into a document having multiple columns. The above processing is initiated by activating one of Print icon 96, Edit icon 97 or Save icon 99, and is similar to the processing described above in the first embodiment, e.g., flattening the document and formatting the document based on the formatting information. Accordingly, a detailed description thereof is omitted for the sake of brevity.

Once the documents whose URLs are stored in the container have been downloaded, formatted according to the preset formats and configurations, and converted into RTF file(s) in step S1011, in step S1012, the RTF file(s) are output. Alternatively, the RTF files(s) can be edited or saved, depending upon which icon on the graphical user interface has been activated.

The invention has been described with respect to particular illustrative embodiments. It is to be understood that the invention is not limited to the above described embodiments and modifications thereto, and that various changes and modifications may be made by those of ordinary skill in the art without departing from the scope of the appended claims as defined in the appended claims.

APPENDIX 1**SAMPLE USER PROFILE**

5 The User Profile is implemented in windows.ini file
format.

10 [Defaults]
Count=4
Title=My Daily Paper

15 [1]
Heading=News In Brief
Site=1
Section=Front Page
MaxLevels=5
MaxPages=10
MaxKBytes=2000
Date=today
Print=level 0
Template=1

20 [2]
Heading=Sports In Brief
Site=2
Section=Sports
Max Levels=0
MaxPages=10
MaxKbytes=200
KeywordFilter="Football" AND "49ers"
Date=today
Print=level 0
Template=1

25 [3]
Heading=Money Matters
Site=1
Section=Business
MaxLevels=1
MaxPages=100
MaxKBytes=20000
KeywordFilter="Computer" OR "hardware" OR
"Software"
Date=today
Print=all
Template=2

30 [4]
Heading=Sri Lanka
Site=3
Section=HotNews
MaxLevels=1
MaxPages=100
MaxKBytes=20000
Date=today
Print=leaves
Template=2

SAMPLE SITE PROFILES

```

5      #Legend:
      #\$W-day of the week
      #\$s-section part of URL

10     [Defaults]
      Count=3

15     [1]
      Title=San Jose Mercury News
      Username=mwickram
      Password=cannon
      StartData=StartHeadlines
      EndData=EndHeadlines
      Home Page=http://www.sjmercury.com/
      SectionURL=http://www.sjmercury.com/\$S.htm
      SectionCount=9
      Section 1=Front Page
      Section 2=International
      Section 3=National
      Section 4=Local & State
      Section 5=Editorials Commentary
      Section 6=Business
      Section 7=Sports
      Section 8=Living
      Section 9=Entertainment

20     [1.Sections]
      Front Page=front
      International=intl
      National=natl
      Local & State=loc
      Editorials & Commentary=edit
      Business=biz
      Sports=spts
      Living=liv
      entertainment=ent

25     [2]
      Title=The San Francisco Chronicle
      Home Page=http://www.sfgate.com/chronicle/
      SectionURL="http://www.sfgate.com/cgi-
                  bin/chronicle/article-
                  list.cgi?\$S:/chronicle/today"
      Section Count=5
      Section 1=News
      Section 2=Business
      Section 3=Sports
      Section 4=Editorial
      Section 5=Datebook

```

50

55

5
[2.Sections]
News=News:MN
Business=Business:BU
Sports=sports:SP
Editorial=Editorial:ED
Datebook=Datebook:DD

10 [3]
Title=The Day News
Home page=
http://www.landa.net/lakehouse/anclWeb/dailynew/
SectionURL="http://www.lanka.net/lakehouse/anclW
15 eb/dailynew/*W/WS.html"
SectionCount =12
Section 1=Business
Section 2=Editorial
Section 3=Features
20 Section 4=Foreign
Section 5=Letters
Section 6=InBrief
Section 7=HotNews
Section 8=Probes
25 Section 9=Military
Section 10=Politics
Section 11=Obituaries
Section 12=Sports

30 [3.Sections]
Business=business/intro
Editorial=editorial/final
Features=features/intro
35 Foreign=foreign/intro
Letters=letters/final
InBrief=inbrief/intro
HotNews=hotnews/intro
Probes=proves/intro
40 Military=military/intro
Politics=politics/intro
Obituaries=obiturai/intro
Sports=sports/intro

45

50

55

APPENDIX 2**SYNTAX FOR RETRIEVAL, EXTRACTION AND PRINTING CRITERIA**

Maximum levels to search: MaxLevels=<#>
 -1: to retrieve all levels
 0-n: to retrieve up to n levels

Maximum pages of the document: MaxPages=<#>
 n: final document not more than n pages

Maximum size of the document: MaxKBytes=<#>
 n: document size not more than n kilo bytes

Exclusion rules:

Date=today|lessthan <#>
 today: retrieve only articles posted today
 lessthan <#>;n: retrieve only articles no
 more than n days old
 Retrieve=all|nosubdir|nothisdir|thissiteonly
 all: allow to fetch pages from other sites
 nosubdir: exclude URLs to subdirectories
 nothisdir: exclude URLs in this directory
 thissiteonly: fetch pages from this site
 only

Keyword search:

KeywordFilter=<keyword> (AND|OR|NOT) <keyword>:
 accumulate only pages containing the
 combination of keywords
 KeywordRank=<#>;n: use fuzzy logic to rank
 pages according to keyword combination in
 KeywordFilter and keep top n ranked pages
 KeywordAuthor=<author>: accumulate only
 pages authored by author
 ExcludeType=ads|nonEnglish
 ads:exclude advertisements
 nonEnglish:exclude articles that are not in
 English

Flattening rules: Print=all|leaves|level=<#>
 all: include all nodes in the tree in the linear
 document
 leaves: include all leaves in the tree in the
 linear document
 level=<#>;n: include up to nth level of the tree
 in the linear document

Formatting rules: Template=<#>
 n: print according to default or user template
 number n

APPENDIX 3

DESCRIPTION OF MODULES

5

Appendix 3A

THE PERSONAL NEWS PROFILE EDITOR MODULE

10

The Profile Editor manages access to the user profiles and is represented by CProfileMgr class. It also manages loading and saving of the profiles. The services provided by Profile Editor are:

15

```

BOOL NewProfile(CString fileName);
    - Creates a new profile given the
      file name.
BOOL OpenProfile();
    - Opens the default profile.
BOOL OpenProfile(CString fileName);
    - Opens the named profile.
CProfileEntry* GetFirstEntry();
    - Loads and returns the next
      profile entry.
CProfileEntry* GetNextEntry();
    - Loads and returns the next
      profile entry.
BOOL WriteEntry(CProfileEntry& entry);
    - Saves a new entry in the profile.

```

20

25

30

35

Each profile entry contains an extraction specification and an output specification as represented by CProfileEntry class. The methods provided are:

40

45

50

55

```

CURL GetsiteId();
    - Returns the site id contained in
      the profile entry.
CExtractionSpec GetExtractionSpec();
    - Returns the extraction
      specification contained in the
      profile entry. Extraction
      specification contains keywords
      for searching, limits for levels,
      pages, size in kilo bytes.
COutputSpec GetOutputSpec();
    - Returns the output specification
      contained in the profile entry.
      Output specification contains
      formatting instructions and tree
      traversal rules.

```

THE Web READER MODULE

5 CWebPage class abstracts the interface to the
 Internet browser and is representative of the actual
 Web page. It will be responsible for fetching a Web
 page, extracting links or references to other URLs
 in the Web page, and maintaining the contents of a
 10 Web page. The methods provided are:

```

    BOOL Load();
      - Fetch the Web page using the URL,
        username and password.
    15        BOOL Parse();
      - Parses the data in the Web page
        and creates a list of links.
        Also resolves the relative URLs
        into absolute URLs.
    20        CURLList* GetLinks();
      - Returns the list of links in the
        Web page.
    CPageData* GetData();
      - Returns the actual text data
        contained in the Web page.
    25        void FilterContent();
      - Extracts title and other
        information according to the site
        data.
    30        CString GetTitle();
      - Returns title and other
        information according to the site
        data.
    35        CString GetAuthor();
      - Returns the author of the Web
        page.
    int GetSize();
      - Returns the size of the data in
        kilo bytes.
  40
  
```

45 CNetwork class will encapsulate OLE functionality
 and provides communication with the Internet
 browser.

```

    CString GetUsername();
      - Determine the currently set
        username.
    50        void SetUsername(LPCTSTR);
      - Set the current username in the
        CNetwork object.
  
```

```

5      Cstring GetPassword();
       - Determine the currently set
         password.
void SetPassword(LPCTSTR);
       - Set the current password in the
         CNetwork object.
10     void Close();
       - Disconnect any active connection
         and reset the CNetwork object.
short Read(BSTR*pBuffer, short iAmount);
       - Read data retrieved by the
         Browser.
15     long GetStatus();
       - Query the status of the current
         load.
BOOL Open(LPCTSTR pURL, short iMethod,
20       LPCTSTR ppostData, long IPostDataSize,
         LPCTSTR pPostHeaders);
       - Initiates the retrieval of a URL
         from the network.
Cstring GetErrorMessage();
25       - Provide the caller with
         internally generated error
         messages.
short GetServerStatus();
       - Determine the error status
         reported by the server.
30     long GetContentType();
       - Return the content length (total
         amount of bytes) of the current
         load.
35     Cstring GetContentEncoding();
       - Return the MIME encoding of the
         current load.
Cstring GetExpires();
40       - Return when the data retrieved by
         this load is no longer considered
         valid.
Cstring Resolve(LPCTSTR pBase, LPCTSTR
45       pRelative);
       - Generate an absolute (fully
         qualified) URL.
BOOL IsFinished();
       - Determine if a load is complete.
50     short BytesReady();
       - Inform the caller of the number
         of bytes prepared to be read.

```

THE SITE DRIVER MODULE

5 The Site Driver will provide the site information to
 the Web Reader. The Site Driver is functionally
 similar to the Profile Editor and is represented by
 CSiteDriver class. Services provided are:

```

 10        BOOL NewProfile(CString fileName);
          - Creates a new profile given the
          file name.
 15        BOOL OpenProfile();
          - Opens the default profile.
 20        BOOL OpenProfile(CString fileName);
          - Opens the named profile.
 25        CSiteProfile* GetFirstSite();
          - Loads and returns the first site
          entry.
 30        CSiteProfile* GetNextSite();
          - Loads and returns the next site
          entry.
 35        BOOL WriteEntry(CSiteProfile& entry);
          - Saves a new entry in the profile.
 40        int NumberOfSites();
          - Returns the number of sites
          specified in the profile.
 45
 50
 55
 60
 65
 70
 75
 80
 85
 90
 95
 100
 105
 110
 115
 120
 125
 130
 135
 140
 145
 150
 155
 160
 165
 170
 175
 180
 185
 190
 195
 200
 205
 210
 215
 220
 225
 230
 235
 240
 245
 250
 255
 260
 265
 270
 275
 280
 285
 290
 295
 300
 305
 310
 315
 320
 325
 330
 335
 340
 345
 350
 355
 360
 365
 370
 375
 380
 385
 390
 395
 400
 405
 410
 415
 420
 425
 430
 435
 440
 445
 450
 455
 460
 465
 470
 475
 480
 485
 490
 495
 500
 505
 510
 515
 520
 525
 530
 535
 540
 545
 550
 555
 560
 565
 570
 575
 580
 585
 590
 595
 600
 605
 610
 615
 620
 625
 630
 635
 640
 645
 650
 655
 660
 665
 670
 675
 680
 685
 690
 695
 700
 705
 710
 715
 720
 725
 730
 735
 740
 745
 750
 755
 760
 765
 770
 775
 780
 785
 790
 795
 800
 805
 810
 815
 820
 825
 830
 835
 840
 845
 850
 855
 860
 865
 870
 875
 880
 885
 890
 895
 900
 905
 910
 915
 920
 925
 930
 935
 940
 945
 950
 955
 960
 965
 970
 975
 980
 985
 990
 995
 1000
 1005
 1010
 1015
 1020
 1025
 1030
 1035
 1040
 1045
 1050
 1055
 1060
 1065
 1070
 1075
 1080
 1085
 1090
 1095
 1100
 1105
 1110
 1115
 1120
 1125
 1130
 1135
 1140
 1145
 1150
 1155
 1160
 1165
 1170
 1175
 1180
 1185
 1190
 1195
 1200
 1205
 1210
 1215
 1220
 1225
 1230
 1235
 1240
 1245
 1250
 1255
 1260
 1265
 1270
 1275
 1280
 1285
 1290
 1295
 1300
 1305
 1310
 1315
 1320
 1325
 1330
 1335
 1340
 1345
 1350
 1355
 1360
 1365
 1370
 1375
 1380
 1385
 1390
 1395
 1400
 1405
 1410
 1415
 1420
 1425
 1430
 1435
 1440
 1445
 1450
 1455
 1460
 1465
 1470
 1475
 1480
 1485
 1490
 1495
 1500
 1505
 1510
 1515
 1520
 1525
 1530
 1535
 1540
 1545
 1550
 1555
 1560
 1565
 1570
 1575
 1580
 1585
 1590
 1595
 1600
 1605
 1610
 1615
 1620
 1625
 1630
 1635
 1640
 1645
 1650
 1655
 1660
 1665
 1670
 1675
 1680
 1685
 1690
 1695
 1700
 1705
 1710
 1715
 1720
 1725
 1730
 1735
 1740
 1745
 1750
 1755
 1760
 1765
 1770
 1775
 1780
 1785
 1790
 1795
 1800
 1805
 1810
 1815
 1820
 1825
 1830
 1835
 1840
 1845
 1850
 1855
 1860
 1865
 1870
 1875
 1880
 1885
 1890
 1895
 1900
 1905
 1910
 1915
 1920
 1925
 1930
 1935
 1940
 1945
 1950
 1955
 1960
 1965
 1970
 1975
 1980
 1985
 1990
 1995
 2000
 2005
 2010
 2015
 2020
 2025
 2030
 2035
 2040
 2045
 2050
 2055
 2060
 2065
 2070
 2075
 2080
 2085
 2090
 2095
 2100
 2105
 2110
 2115
 2120
 2125
 2130
 2135
 2140
 2145
 2150
 2155
 2160
 2165
 2170
 2175
 2180
 2185
 2190
 2195
 2200
 2205
 2210
 2215
 2220
 2225
 2230
 2235
 2240
 2245
 2250
 2255
 2260
 2265
 2270
 2275
 2280
 2285
 2290
 2295
 2300
 2305
 2310
 2315
 2320
 2325
 2330
 2335
 2340
 2345
 2350
 2355
 2360
 2365
 2370
 2375
 2380
 2385
 2390
 2395
 2400
 2405
 2410
 2415
 2420
 2425
 2430
 2435
 2440
 2445
 2450
 2455
 2460
 2465
 2470
 2475
 2480
 2485
 2490
 2495
 2500
 2505
 2510
 2515
 2520
 2525
 2530
 2535
 2540
 2545
 2550
 2555
 2560
 2565
 2570
 2575
 2580
 2585
 2590
 2595
 2600
 2605
 2610
 2615
 2620
 2625
 2630
 2635
 2640
 2645
 2650
 2655
 2660
 2665
 2670
 2675
 2680
 2685
 2690
 2695
 2700
 2705
 2710
 2715
 2720
 2725
 2730
 2735
 2740
 2745
 2750
 2755
 2760
 2765
 2770
 2775
 2780
 2785
 2790
 2795
 2800
 2805
 2810
 2815
 2820
 2825
 2830
 2835
 2840
 2845
 2850
 2855
 2860
 2865
 2870
 2875
 2880
 2885
 2890
 2895
 2900
 2905
 2910
 2915
 2920
 2925
 2930
 2935
 2940
 2945
 2950
 2955
 2960
 2965
 2970
 2975
 2980
 2985
 2990
 2995
 3000
 3005
 3010
 3015
 3020
 3025
 3030
 3035
 3040
 3045
 3050
 3055
 3060
 3065
 3070
 3075
 3080
 3085
 3090
 3095
 3100
 3105
 3110
 3115
 3120
 3125
 3130
 3135
 3140
 3145
 3150
 3155
 3160
 3165
 3170
 3175
 3180
 3185
 3190
 3195
 3200
 3205
 3210
 3215
 3220
 3225
 3230
 3235
 3240
 3245
 3250
 3255
 3260
 3265
 3270
 3275
 3280
 3285
 3290
 3295
 3300
 3305
 3310
 3315
 3320
 3325
 3330
 3335
 3340
 3345
 3350
 3355
 3360
 3365
 3370
 3375
 3380
 3385
 3390
 3395
 3400
 3405
 3410
 3415
 3420
 3425
 3430
 3435
 3440
 3445
 3450
 3455
 3460
 3465
 3470
 3475
 3480
 3485
 3490
 3495
 3500
 3505
 3510
 3515
 3520
 3525
 3530
 3535
 3540
 3545
 3550
 3555
 3560
 3565
 3570
 3575
 3580
 3585
 3590
 3595
 3600
 3605
 3610
 3615
 3620
 3625
 3630
 3635
 3640
 3645
 3650
 3655
 3660
 3665
 3670
 3675
 3680
 3685
 3690
 3695
 3700
 3705
 3710
 3715
 3720
 3725
 3730
 3735
 3740
 3745
 3750
 3755
 3760
 3765
 3770
 3775
 3780
 3785
 3790
 3795
 3800
 3805
 3810
 3815
 3820
 3825
 3830
 3835
 3840
 3845
 3850
 3855
 3860
 3865
 3870
 3875
 3880
 3885
 3890
 3895
 3900
 3905
 3910
 3915
 3920
 3925
 3930
 3935
 3940
 3945
 3950
 3955
 3960
 3965
 3970
 3975
 3980
 3985
 3990
 3995
 4000
 4005
 4010
 4015
 4020
 4025
 4030
 4035
 4040
 4045
 4050
 4055
 4060
 4065
 4070
 4075
 4080
 4085
 4090
 4095
 4100
 4105
 4110
 4115
 4120
 4125
 4130
 4135
 4140
 4145
 4150
 4155
 4160
 4165
 4170
 4175
 4180
 4185
 4190
 4195
 4200
 4205
 4210
 4215
 4220
 4225
 4230
 4235
 4240
 4245
 4250
 4255
 4260
 4265
 4270
 4275
 4280
 4285
 4290
 4295
 4300
 4305
 4310
 4315
 4320
 4325
 4330
 4335
 4340
 4345
 4350
 4355
 4360
 4365
 4370
 4375
 4380
 4385
 4390
 4395
 4400
 4405
 4410
 4415
 4420
 4425
 4430
 4435
 4440
 4445
 4450
 4455
 4460
 4465
 4470
 4475
 4480
 4485
 4490
 4495
 4500
 4505
 4510
 4515
 4520
 4525
 4530
 4535
 4540
 4545
 4550
 4555
 4560
 4565
 4570
 4575
 4580
 4585
 4590
 4595
 4600
 4605
 4610
 4615
 4620
 4625
 4630
 4635
 4640
 4645
 4650
 4655
 4660
 4665
 4670
 4675
 4680
 4685
 4690
 4695
 4700
 4705
 4710
 4715
 4720
 4725
 4730
 4735
 4740
 4745
 4750
 4755
 4760
 4765
 4770
 4775
 4780
 4785
 4790
 4795
 4800
 4805
 4810
 4815
 4820
 4825
 4830
 4835
 4840
 4845
 4850
 4855
 4860
 4865
 4870
 4875
 4880
 4885
 4890
 4895
 4900
 4905
 4910
 4915
 4920
 4925
 4930
 4935
 4940
 4945
 4950
 4955
 4960
 4965
 4970
 4975
 4980
 4985
 4990
 4995
 5000
 5005
 5010
 5015
 5020
 5025
 5030
 5035
 5040
 5045
 5050
 5055
 5060
 5065
 5070
 5075
 5080
 5085
 5090
 5095
 5100
 5105
 5110
 5115
 5120
 5125
 5130
 5135
 5140
 5145
 5150
 5155
 5160
 5165
 5170
 5175
 5180
 5185
 5190
 5195
 5200
 5205
 5210
 5215
 5220
 5225
 5230
 5235
 5240
 5245
 5250
 5255
 5260
 5265
 5270
 5275
 5280
 5285
 5290
 5295
 5300
 5305
 5310
 5315
 5320
 5325
 5330
 5335
 5340
 5345
 5350
 5355
 5360
 5365
 5370
 5375
 5380
 5385
 5390
 5395
 5400
 5405
 5410
 5415
 5420
 5425
 5430
 5435
 5440
 5445
 5450
 5455
 5460
 5465
 5470
 5475
 5480
 5485
 5490
 5495
 5500
 5505
 5510
 5515
 5520
 5525
 5530
 5535
 5540
 5545
 5550
 5555
 5560
 5565
 5570
 5575
 5580
 5585
 5590
 5595
 5600
 5605
 5610
 5615
 5620
 5625
 5630
 5635
 5640
 5645
 5650
 5655
 5660
 5665
 5670
 5675
 5680
 5685
 5690
 5695
 5700
 5705
 5710
 5715
 5720
 5725
 5730
 5735
 5740
 5745
 5750
 5755
 5760
 5765
 5770
 5775
 5780
 5785
 5790
 5795
 5800
 5805
 5810
 5815
 5820
 5825
 5830
 5835
 5840
 5845
 5850
 5855
 5860
 5865
 5870
 5875
 5880
 5885
 5890
 5895
 5900
 5905
 5910
 5915
 5920
 5925
 5930
 5935
 5940
 5945
 5950
 5955
 5960
 5965
 5970
 5975
 5980
 5985
 5990
 5995
 6000
 6005
 6010
 6015
 6020
 6025
 6030
 6035
 6040
 6045
 6050
 6055
 6060
 6065
 6070
 6075
 6080
 6085
 6090
 6095
 6100
 6105
 6110
 6115
 6120
 6125
 6130
 6135
 6140
 6145
 6150
 6155
 6160
 6165
 6170
 6175
 6180
 6185
 6190
 6195
 6200
 6205
 6210
 6215
 6220
 6225
 6230
 6235
 6240
 6245
 6250
 6255
 6260
 6265
 6270
 6275
 6280
 6285
 6290
 6295
 6300
 6305
 6310
 6315
 6320
 6325
 6330
 6335
 6340
 6345
 6350
 6355
 6360
 6365
 6370
 6375
 6380
 6385
 6390
 6395
 6400
 6405
 6410
 6415
 6420
 6425
 6430
 6435
 6440
 6445
 6450
 6455
 6460
 6465
 6470
 6475
 6480
 6485
 6490
 6495
 6500
 6505
 6510
 6515
 6520
 6525
 6530
 6535
 6540
 6545
 6550
 6555
 6560
 6565
 6570
 6575
 6580
 6585
 6590
 6595
 6600
 6605
 6610
 6615
 6620
 6625
 6630
 6635
 6640
 6645
 6650
 6655
 6660
 6665
 6670
 6675
 6680
 6685
 6690
 6695
 6700
 6705
 6710
 6715
 6720
 6725
 6730
 6735
 6740
 6745
 6750
 6755
 6760
 6765
 6770
 6775
 6780
 6785
 6790
 6795
 6800
 6805
 6810
 6815
 6820
 6825
 6830
 6835
 6840
 6845
 6850
 6855
 6860
 6865
 6870
 6875
 6880
 6885
 6890
 6895
 6900
 6905
 6910
 6915
 6920
 6925
 6930
 6935
 6940
 6945
 6950
 6955
 6960
 6965
 6970
 6975
 6980
 6985
 6990
 6995
 7000
 7005
 7010
 7015
 7020
 7025
 7030
 7035
 7040
 7045
 7050
 7055
 7060
 7065
 7070
 7075
 7080
 7085
 7090
 7095
 7100
 7105
 7110
 7115
 7120
 7125
 7130
 7135
 7140
 7145
 7150
 7155
 7160
 7165
 7170
 7175
 7180
 7185
 7190
 7195
 7200
 7205
 7210
 7215
 7220
 7225
 7230
 7235
 7240
 7245
 7250
 7255
 7260
 7265
 7270
 7275
 7280
 7285
 7290
 7295
 7300
 7305
 7310
 7315
 7320
 7325
 7330
 7335
 7340
 7345
 7350
 7355
 7360
 7365
 7370
 7375
 7380
 7385
 7390
 7395
 7400
 7405
 7410
 7415
 7420
 7425
 7430
 7435
 7440
 7445
 7450
 7455
 7460
 7465
 7470
 7475
 7480
 7485
 7490
 7495
 7500
 7505
 7510
 7515
 7520
 7525
 7530
 7535
 7540
 7545
 7550
 7555
 7560
 7565
 7570
 7575
 7580
 7585
 7590
 7595
 7600
 7605
 7610
 7615
 7620
 7625
 7630
 7635
 7640
 7645
 7650
 7655
 7660
 7665
 7670
 7675
 7680
 7685
 7690
 7695
 7700
 7705
 7710
 7715
 7720
 7725
 7730
 7735
 7740
 7745
 7750
 7755
 7760
 7765
 7770
 7775
 7780
 7785
 7790
 7795
 7800
 7805
 7810
 7815
 7820
 7825
 7830
 7835
 7840
 7845
 7850
 7855
 7860
 7865
 7870
 7875
 7880
 7885
 7890
 7895
 7900
 7905
 7910
 7915
 7920
 7925
 7930
 7935
 7940
 7945
 7950
 7955
 7960
 7965
 7970
 7975
 7980
 7985
 7990
 7995
 8000
 8005
 8010
 8015
 8020
 8025
 8030
 8035
 8040
 8045
 8050
 8055
 8060
 8065
 8070
 8075
 8080
 8085
 8090
 8095
 8100
 8105
 8110
 8115
 8120
 8125
 8130
 8135
 8140
 8145
 8150
 8155
 8160
 8165
 8170
 8175
 8180
 8185
 8190
 8195
 8200
 8205
 8210
 8215
 8220
 8225
 8230
 8235
 8240
 8245
 8250
 8255
 8260
 8265
 8270
 8275
 8280
 8285
 8290
 8295
 8300
 8305
 8310
 8315
 8320
 8325
 8330
 8335
 8340
 8345
 8350
 8355
 8360
 8365
 8370
 8375
 8380
 8385
 8390
 8395
 8400
 8405
 8410
 8415
 8420
 8425
 8430
 8435
 8440
 8445
 8450
 8455
 8460
 8465
 8470
 8475
 8480
 8485
 8490
 8495
 8500
 8505
 8510
 8515
 8520
 8525
 8530
 8535
 8540
 8545
 8550
 8555
 8560
 8565
 8570
 8575
 8580
 8585
 8590
 8595
 8600
 8605
 8610
 8615
 8620
 8625
 8630
 8635
 8640
 8645
 8650
 8655
 8660
 8665
 8670
 8675
 8680
 8685
 8690
 8695
 8700
 8705
 8710
 8715
 8720
 8725
 8730
 8735
 8740
 8745
 8750
 8755
 8760
 8765
 8770
 8775
 8780
 8785
 8790
 8795
 8800
 8805
 8810
 8815
 8820
 8825
 8830
 8835
 8840
 8845
 8850
 8855
 8860
 8865
 8870
 8875
 8880
 8885
 8890
 8895
 8900
 8905
 8910
 8915
 8920
 8925
 8930
 8935
 8940
 8945
 8950
 8955
 8960
 8965
 8970
 8975
 8980
 8985
 8990
 8995
 9000
 9005
 9010
 9015
 9020
 9025
 9030
 9035
 9040
 9045
 9050
 9055
 9060
 9065
 9070
 9075
 9080
 9085
 9090
 9095
 9100
 9105
 9110
 9115
 9120
 9125
 9130
 9135
 9140
 9145
 9150
 9155
 9160
 9165
 9170
 9175
 9180
 9185
 9190
 9195
 9200
 9205
 9210
 9215
 9220
 9225
 9230
 9235
 9240
 9245
 9250
 9255
 9260
 9265
 9270
 9275
 9280
 9285
 9290
 9295
 9300
 9305
 9310
 9315
 9320
 9325
 9330
 9335
 9340
 9345
 9350
 9355
 9360
 9365
 9370
 9375
 9380
 9385

```

Appendix 3B

TREE MANAGER MODULE

5

10

Tree Manager will maintain the most central data structure in this program, which is a tree of Web page nodes and is represented by the CPageTree. CPageTree will traverse the WWW to retrieve the necessary Web pages according to the extraction specification and builds the tree. The methods provided are:

15

```
CPageTreeNode* GetRoot();
    - Returns the root node of the
      tree.
BOOL Build(CURL URL, CExtractionSpec&
spec);
    - Builds the tree according to the
      personal news profile extraction
      specification.
```

20

25

Each node in the page tree is represented by a CPageTreeNode. Methods provided are:

30

```
BOOL AddChild(CWebPage* page);
    - Adds a child node with Web page
      data.
CWebPage* GetPage();
    - Returns the Web page contained in
      the node.
int NumberOfChildren();
    - Returns the number of children
      belonging to the node.
BOOL IsLeaf();
    - Returns TRUE if a leaf node,
      i.e., no children.
```

35

40

45

To traverse the Web page tree, a CTreeIterator class is defined with different traversal methods. Methods provided are:

50

```
void Reset();
    - Cancels the current transferral,
      and initializes state data.
CPageTreeNode* GetNextNode();
    - Returns the next node in the tree
      in a depth first search.
```

55

```

5      CPage TreeNode* GetNextSibling();
      - Returns the next node in the tree
      in a breadth first search.
CPage TreeNode* GetNextLeaf();
      - Returns the next leaf in the tree
      in a depth first search.

```

10 THE FORMATTER MODULE

15 Input to this module will be the Web page tree
 created by the Tree Manager and the output
 specification contained in the user profile.
 20 Formatter will traverse the tree according to the
 rules specified in the output specification and the
 final document will be formatted using the
 formatting instructions in the output specification
 and the formatting contained in the Web pages such
 as headings, paragraphs and lists etc.

25 The output document will be in Rich Text Format
 (RTF) and will be accessible by many applications.
 RTF is a advanced formatting language for text,
 providing document, section and paragraph
 formatting, style sheets, headers and footers, and
 with support for Uniconde. Image formats supported
 are DIB, DDB, WMF, OS/2 metafiles. There is no
 support for Web images which are of the GIF format.
 30 A third party library will need to be purchased in
 order to do the conversion of the GIF to DIB format
 or one can be developed in-house.

35 The prototype creates a HTML file as the output.

The formatter is represented by the CFormatter
 class. The methods provided are:

```

40      BOOL OpenHTMLFile(CString fileName);
      - Opens the named HTML file for
      output.
void CloseHTMLFile();
      - Closes and saves the HTML file.
BOOL PrintHTML(CPageTree& root,
COutputSpec& format);
      - Given the root and the output
      specification, traverses the tree
      and prints the contents in the
      Web pages in HTML format.
BOOL OpenRTFFile(CString fileName);
      - Opens the named RTF file for
      output.

```

5 void CloseRTFFile();
 - Closes and saves the RTF file.
 10 BOOL PrintRTF(CPageTree& root, COutputSpec&
 format);
 - Given the root and the output
 specification, traverses the tree
 and prints the contents in the
 Web pages in RTF format.

15 BOOL Print(CPageTree& root, COutputSpec& format);
 - Given the root and the output specification
 at, traverses the tree and prints the
 contents in the Web pages to the default
 printer.

20

Claims

1. A method for formatting data from at least one hypermedia document, comprising the steps of:

25 an accessing step to access the at least one hypermedia document;
 a retrieving step to retrieve data from the hypermedia document into an extracted data tree, wherein the data
 is retrieved based on a structure of the hypermedia document;
 a flattening step to flatten the extracted data tree into a linear document; and
 30 a formatting step to format the linear document into a formatted document.

2. The method of Claim 1, further comprising the step of printing the formatted document.
3. The method of Claim 1, wherein said hypermedia document is located on the World Wide Web.
- 35 4. The method of Claim 1, wherein said hypermedia document is located on the Internet.
5. The method of Claim 1, wherein said hypermedia document is located on an intranet.
- 40 6. The method of Claim 1, wherein said accessing step, said retrieving step, said flattening step, and said formatting
 step are performed in accordance with a personal-news-profile.
7. A method of creating a personal-news-profile for retrieving data from a hypermedia-linked computer network, comprising the steps of:

- 45 accessing the hypermedia-linked computer network;
 entering a learning mode;
 traversing sites on the hypermedia-linked computer network with commands;
 extracting at least one rule from the commands; and
 50 compiling the at least one rule into the personal-news-profile.
8. The method of Claim 7, wherein the at least one rule specifies structural characteristics of sites for traversing the hypermedia-linked computer network.
- 55 9. The method of Claim 8, wherein the at least one rule also specifies content-based criteria for traversing the hypermedia-linked computer network.
10. A personalization system for creating a personalization profile for a Web site retrieval data retrieval system, the per-

sonalization system comprising:

- an input device for inputting data and commands to access the World Wide Web;
- a connection to the World Wide Web;
- 5 a memory for storing a Web reader, the Web reader for accessing the World Wide Web via the connection to the World Wide Web according to commands from the personalization system; and
- a processor for launching the personalization system in response to a user command, wherein the personalization system, upon being launched, (1) launches the Web reader, (2) accesses the World Wide Web via the Web reader, (3) enters a learning mode, (4) sends commands to the Web reader to traverse the World Wide Web according to user commands, (5) extracts at least one rule from the user commands, (6) compiles the at least one rule into a personalization profile, and (7) stores the personalization profile.

10 11. A method for retrieving articles from a hypermedia-linked computer network and for formatting the articles into a personalized newspaper, the method comprising the steps of:

- 15 retrieving a stored personal-news-profile which comprises address data for a site on the hypermedia-linked computer network, command data for accessing data from the site, and newspaper layout commands;
- contacting the site based on address data stored in the personal-news-profile;
- downloading articles from the site based on command data stored in the personal-news-profile;
- 20 flattening the articles into a linear document; and
- formatting the linear document into the personalized newspaper according to layout commands stored in the personal-news-profile.

25 12. The method of Claim 11, further comprising the step of printing the personalized newspaper.

13. The method of Claim 11, wherein said hypermedia-linked computer network is the World Wide Web.

14. The method of Claim 11, wherein said hypermedia-linked computer network is on the Internet.

30 15. The method of Claim 11, wherein said hypermedia-linked computer network is on an intranet.

16. The method of Claim 11, wherein the command data for accessing data includes data for selecting articles based on a structure of the site.

35 17. The method of Claim 16, wherein the command data for accessing data also includes data for selecting articles based on a content of the articles.

18. A World Wide Web site data retrieval system for accessing at least one Web site, for retrieving data from the Web site, and for formatting the data into a personalized document, the system comprising:

- 40 an input device for inputting data and commands to access the World Wide Web;
- a memory for storing a Web site data retrieval driver which includes a Web reader, stored Web site address information, stored Web site commands, and stored format information, wherein the memory also includes process steps to connect to a Web site and to issue commands within the connected Web site;
- 45 a connection to the World Wide Web; and
- a processor for launching the Web site data retrieval driver in response to a user inputting a command to access the World Wide Web, wherein the Web site retrieval driver, upon being launched, (1) launches the Web reader to connect to the World Wide Web via said connection, (2) retrieves the Web site address information and Web site commands, (3) instructs the Web reader to access the Web site based on the Web site address information and Web site commands, (4) downloads Web site data from the Web site based on the Web site commands, wherein the data is downloaded with reference to a linked list so as to avoid hypermedia-links that form loops and so as to avoid repetitive downloading of data that has already been downloaded, (5) stores the Web site data in a linear document, (6) repeats steps 1 through 5 until all addresses in the stored Web site address information have been accessed, and (7) formats the linear document into the personalized document based on the format information.

55 19. The Web site data retrieval system of Claim 18, wherein the Web site address information, the Web site commands, and the format information stored in the memory form a personalized-news-profile.

20. The Web site data retrieval system of Claim 18, further comprising a printer for printing the personalized document.
21. The Web site data retrieval system of Claim 18, wherein the personalized document represents a personalized newspaper.
- 5 22. The Web site data retrieval system of Claim 18, wherein the personalized document represents a personalized magazine.
- 10 23. The Web site data retrieval system of Claim 18, wherein the personalized document represents a personalized book.
- 15 24. Computer executable process steps stored on a computer-readable medium, said steps for accessing World Wide Web sites for retrieving data at the sites and for formatting the data into a personalized document, said steps comprising:
 - a connecting step to connect to the World Wide Web;
 - a retrieving step to retrieve user-defined Web site address information, user-defined Web site commands, and user-defined formatting commands;
 - 20 an activating step to activate a Web reader so as to access a Web site based on the user-defined Web site address information and retrieving data from within the Web site based on the user-defined Web site commands;
 - a downloading step to download the retrieved Web site data from the accessed Web site into an extracted data tree;
 - 25 a flattening step to flatten the extracted data tree into a linear document;
 - a step to repeat the downloading step and the flattening step until all addresses in the user-defined Web site address information have been accessed; and
 - a formatting step to format the stored data into the personalized document based on the user-defined formatting commands.
- 30 25. The computer executable process steps of Claim 24, further comprising a spooling step to spool the personalized document to an output device.
26. The computer executable process steps of Claim 25, wherein the output device is a printer.
- 35 27. The computer executable process steps of Claim 25, wherein the output device is a display.
28. The computer executable process steps of Claim 24, wherein the user-defined Web site commands include commands for selecting data based on a structure of the Web site.
- 40 29. The computer executable process steps of Claim 28, wherein the user-defined Web site commands also include commands for selecting data based on a content of the Web site.
30. An apparatus for retrieving news articles from on-line news services on the World Wide Web and formatting the news articles into a personalized newspaper, the apparatus comprising:
 - 45 first storage means for storing (1) a personal-news-profile which comprises addresses data and command data for accessing data from a Web site, and (2) newspaper format commands;
 - retrieval means for retrieving the stored personal-news-profile and accessing data stored therein;
 - activating means for activating a Web reader to contact a Web site based on address data stored in the personal-news-profile;
 - 50 downloading means for downloading news articles from the contacted Web site based on command data stored in the personal-news-profile;
 - second storage means for storing the downloaded news articles; and
 - 55 formatting means for formatting the stored news articles into the personalized newspaper based on the newspaper format commands stored in the personal-news-profile.
31. The apparatus of Claim 30, further comprising spooling means for spooling the personalized newspaper to a printer.

32. A method for formatting data from a hypermedia document into a personalized document, comprising the steps of:

- 5 a location specifying step to specify a location of the hypermedia document;
- a type specifying step to specify the type of the hypermedia document;
- a scope specifying step to specify the scope of data to retrieve from the hypermedia document, wherein the scope is based on a structure of the hypermedia document;
- a format specifying step to specify a format for formatting the data retrieved from the hypermedia document into the personalized document;
- 10 an accessing step to access the hypermedia document found at the location specified in the location specifying step;
- a retrieving step to retrieve data from the hypermedia document accessed in the accessing step, wherein the data is retrieved in accordance with the type specified in the type specifying step and in accordance with the scope specified in the scope specifying step; and
- 15 a formatting step to format the data retrieved in the retrieving step into the personalized document, wherein the data is formatted in accordance with the format specified in the format specifying step.

33. The method of Claim 32, further comprising a printing step to print the personalized document.

34. The method of Claim 32, wherein the location specified in the location specifying step is a filename.

20 35. The method of Claim 32, wherein the location specified in the location specifying step is a uniform resource locator for the World Wide Web.

25 36. A method of processing a hypermedia document, comprising the steps of:

- 25 accessing the hypermedia document;
- extracting addresses from the hypermedia document;
- storing the addresses extracted from the hypermedia document in a container in a memory;
- activating a processing function to process data stored at the addresses stored in the container;
- 30 downloading the data stored at the addresses in the container into the memory;
- extracting predetermined data from downloaded data in accordance with predetermined configuration information;
- formatting the predetermined data in accordance with predefined formatting settings to generate a formatted document; and
- 35 processing the formatted document in accordance with the processing function.

37. A method according to Claim 36, further comprising a step of previewing the formatted document prior to processing the formatted document.

40 38. A method according to Claim 37, further comprising the steps of:

- 40 changing the formatting settings after previewing the document and before processing the formatted document in accordance with the processing function;
- re-activating the processing function; and
- 45 re-formatting the data in accordance with changed formatting settings to generate the formatted document.

39. A method according to Claim 36, wherein the addresses are stored in the container in the order that the addresses are input into the container; and

50 wherein the processing function processes the predetermined data in the order that the addresses are stored in the container.

40. A method according to Claim 39, further comprising the step of rearranging the addresses stored in the container by dragging and dropping the addresses within the container.

55 41. A method according to Claim 36, further comprising a step of inputting the formatting settings and configuration information via a graphical user interface.

42. A method according to Claim 41, wherein the graphical user interface comprises plural processing icons, one of

which activates the processing function.

- 43. A method according to Claim 42, wherein the graphical user interface is displayed in plural modes.
- 5 44. A method according to Claim 43, wherein the plural modes comprise (1) a fully-functional mode in which the graphical user interface displays formatting fields, processing options, menus and the processing icons, and (2) a minimizing mode in which the graphical user interface displays only the processing icons.
- 10 45. A method according to Claim 44, wherein the graphical user interface displayed in the minimizing mode is displayed during browsing the hypermedia document.
- 46. An apparatus for processing a hypermedia document, comprising:
 - 15 a Web read which accesses the hypermedia document;
 - means for extracting addresses from the hypermedia document;
 - a memory including a container which stores the addresses extracted from the hypermedia document;
 - a graphical user interface having processing icons which activate at least one processing function to process data stored at the addresses stored in the container; and
 - 20 processing means which (1) downloads the data stored at the addresses stored in the container into the memory, (2) extracts predetermined data from downloaded data in accordance with predefined configuration settings, (3) formats the predetermined data in accordance with predefined formatting settings to generate a formatted document, and (4) processes the formatted document in accordance with the processing function.
- 47. An apparatus according to Claim 46, further comprising previewing means for previewing the formatted document prior to processing the formatted document.
- 25 48. An apparatus according to Claim 46, wherein the addresses are stored in the container in the order that the addresses are input into the container; and
 - 30 wherein the processing function processes the predetermined data in the order that the addresses are stored in the container.
- 49. An apparatus according to Claim 48, further comprising dragging and dropping means for dragging and dropping the addresses listed in the container in order to rearrange the addresses in the container.
- 35 50. An apparatus according to Claim 46, further comprising inputting means for inputting the formatting settings and configuration information via a graphical user interface.
- 51. An apparatus according to Claim 50, wherein the graphical user interface comprises plural processing icons, one of which activates to the processing function.
- 40 52. An apparatus according to Claim 51, further comprising display means for displaying the graphical user interface in plural modes.
- 53. An apparatus according to Claim 52, wherein the plural modes comprise (1) a fully-functional mode in which the graphical user interface displays formatting fields, processing options, menus and the processing icons, and (2) a minimizing mode in which the graphical user interface displays only the processing icons.
- 45 54. An apparatus according to Claim 53, wherein the graphical user interface displayed in the minimizing mode is displayed during browsing the hypermedia document.

50

55

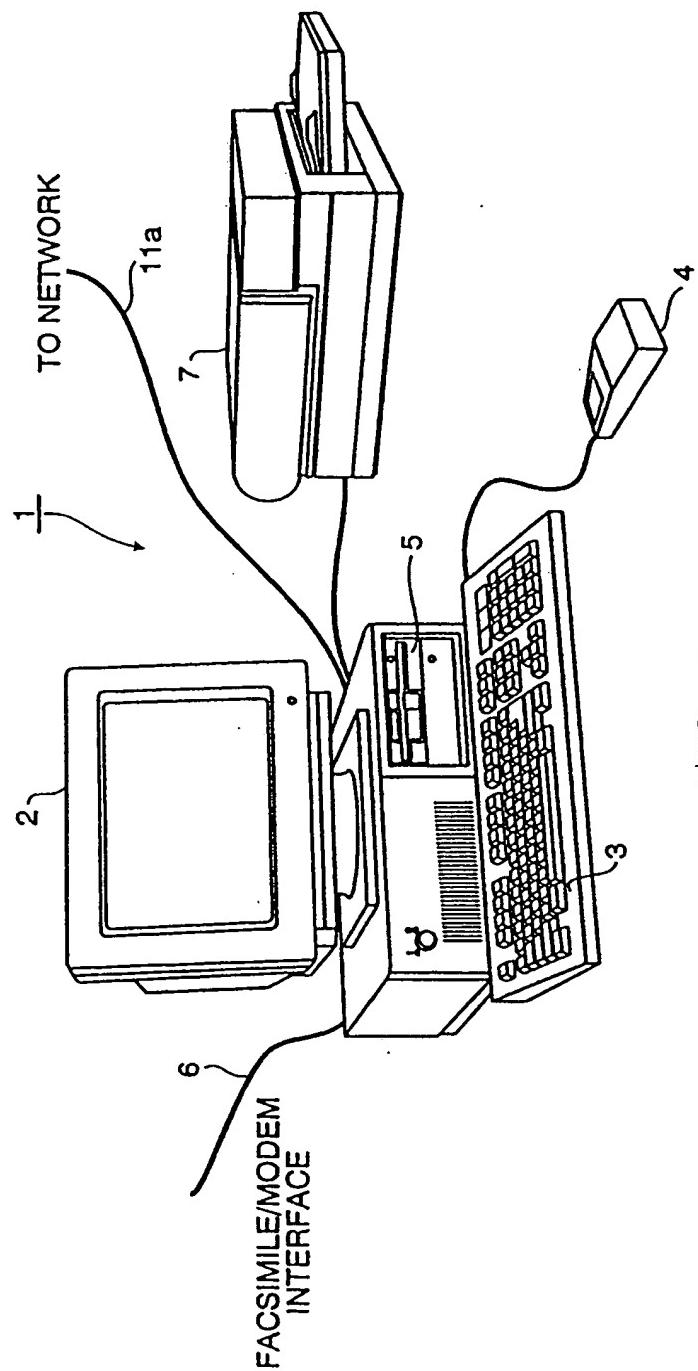


FIG. 1

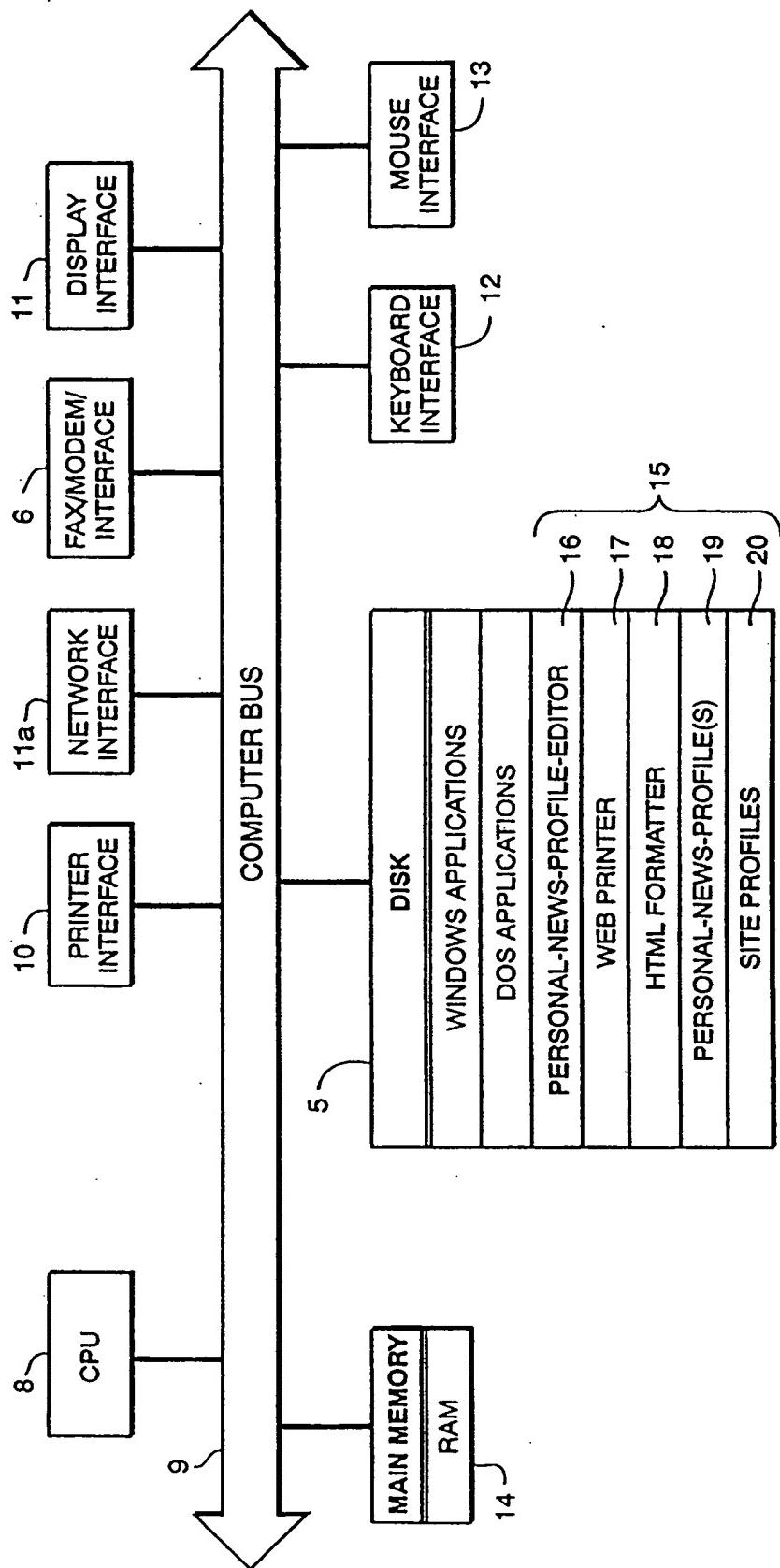
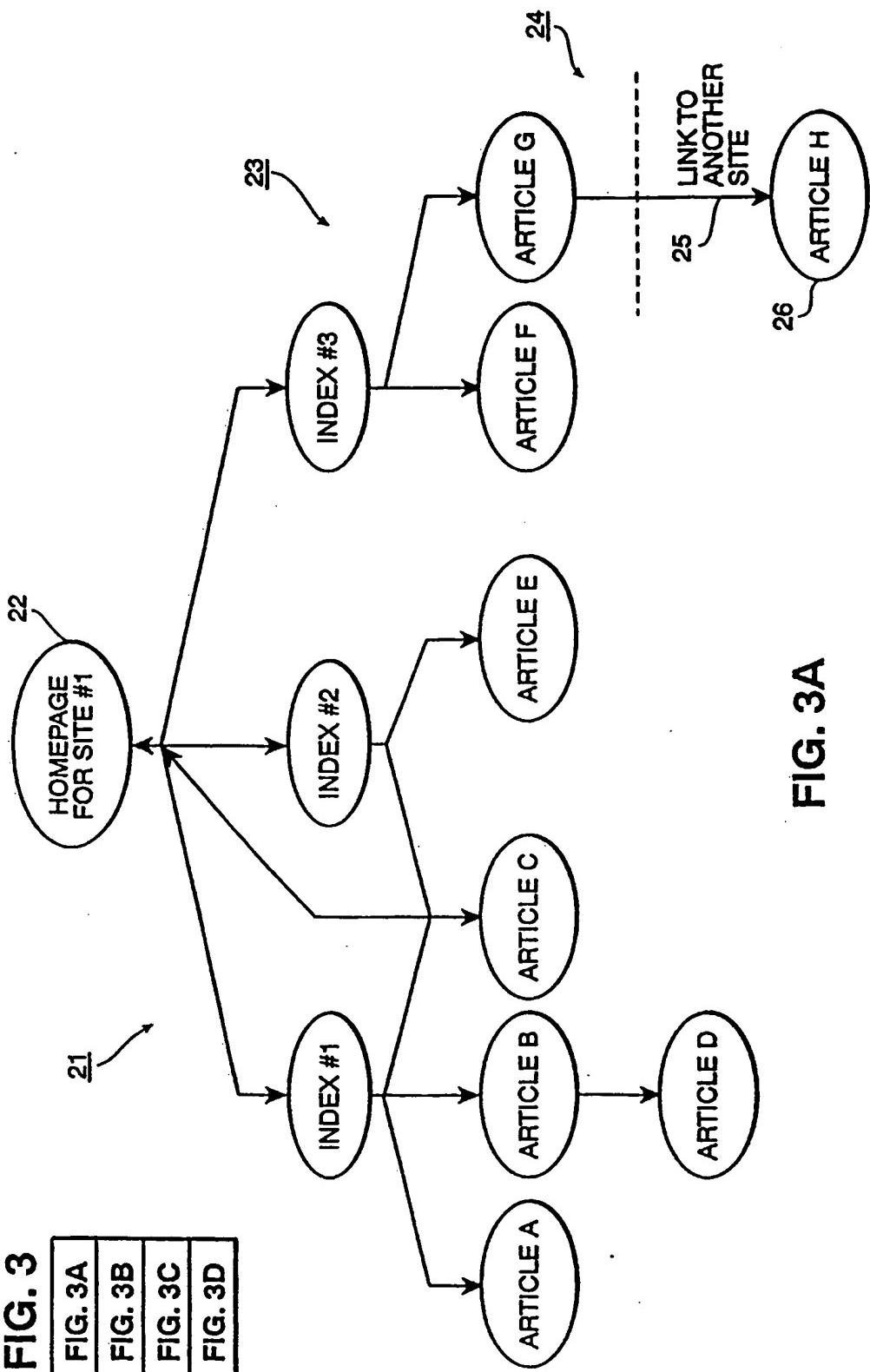
**FIG. 2**

FIG. 3

FIG. 3A
FIG. 3B
FIG. 3C
FIG. 3D

**FIG. 3A**

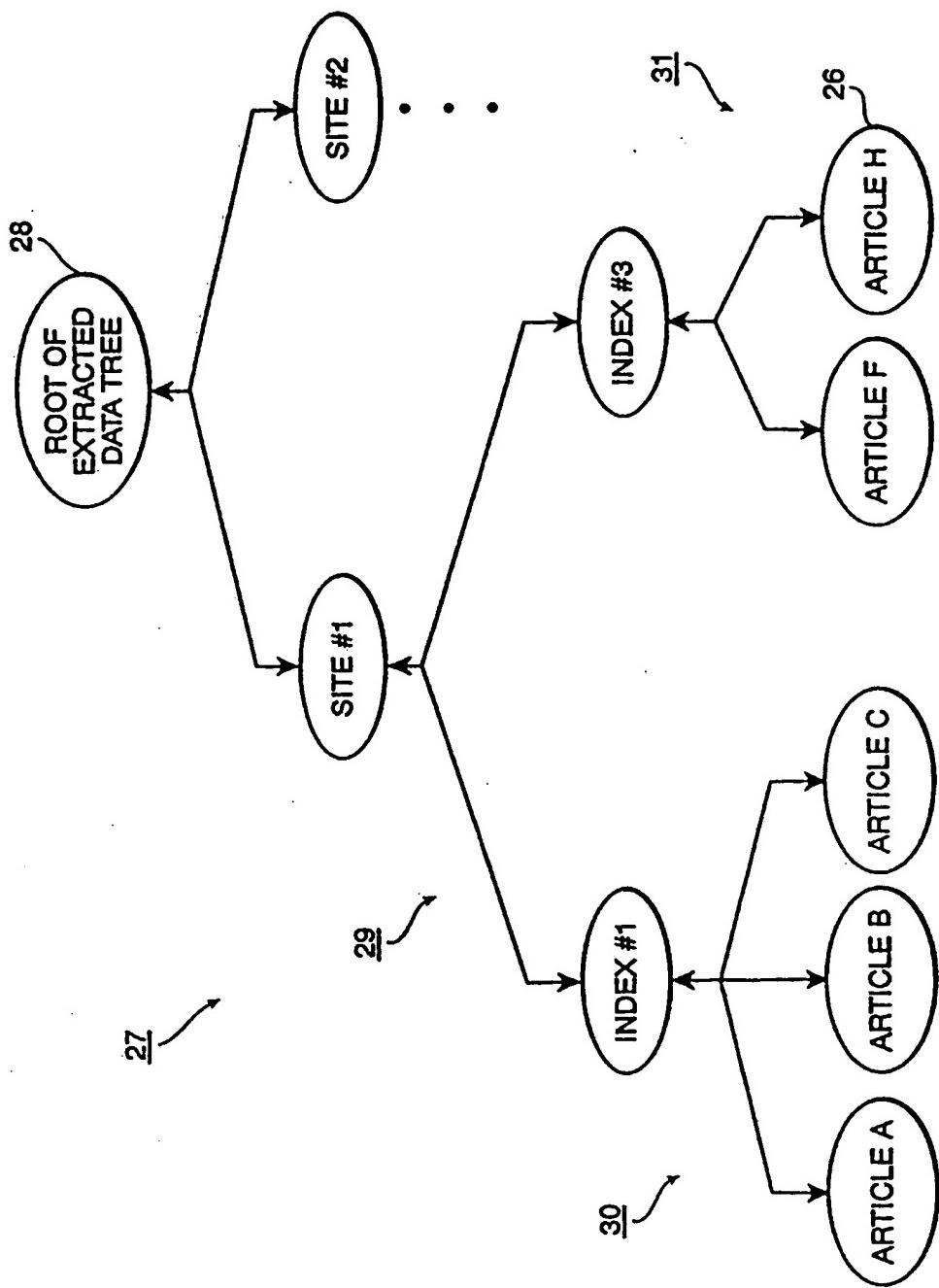
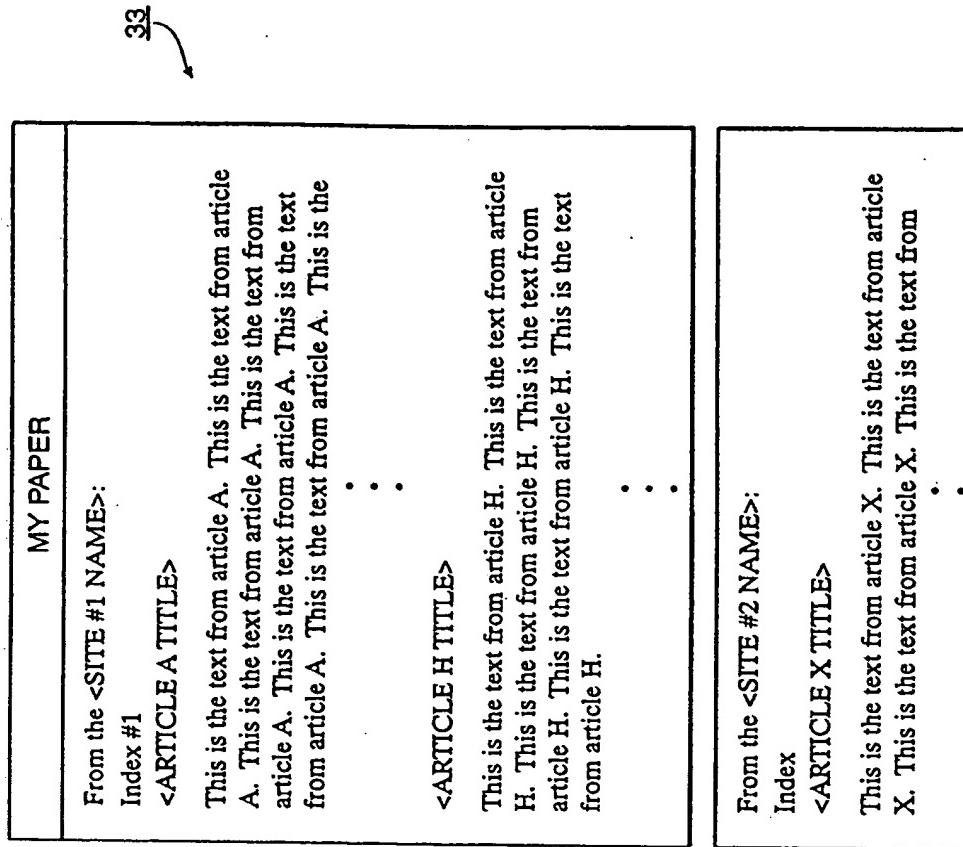
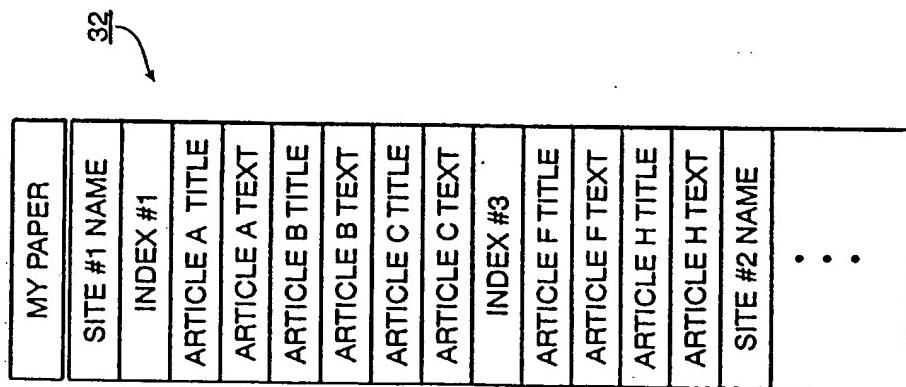


FIG. 3B

**FIG. 3D****FIG. 3C**

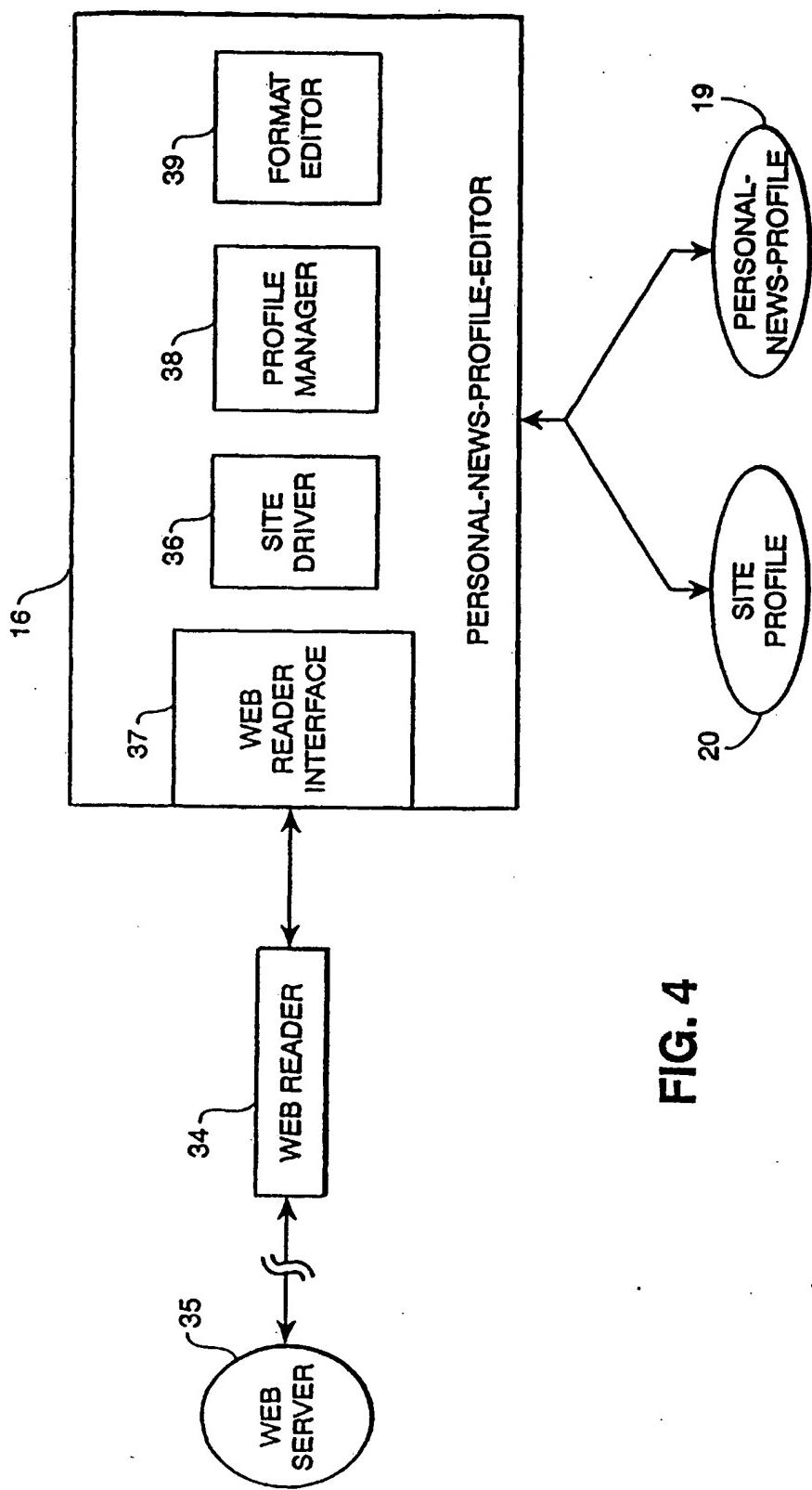
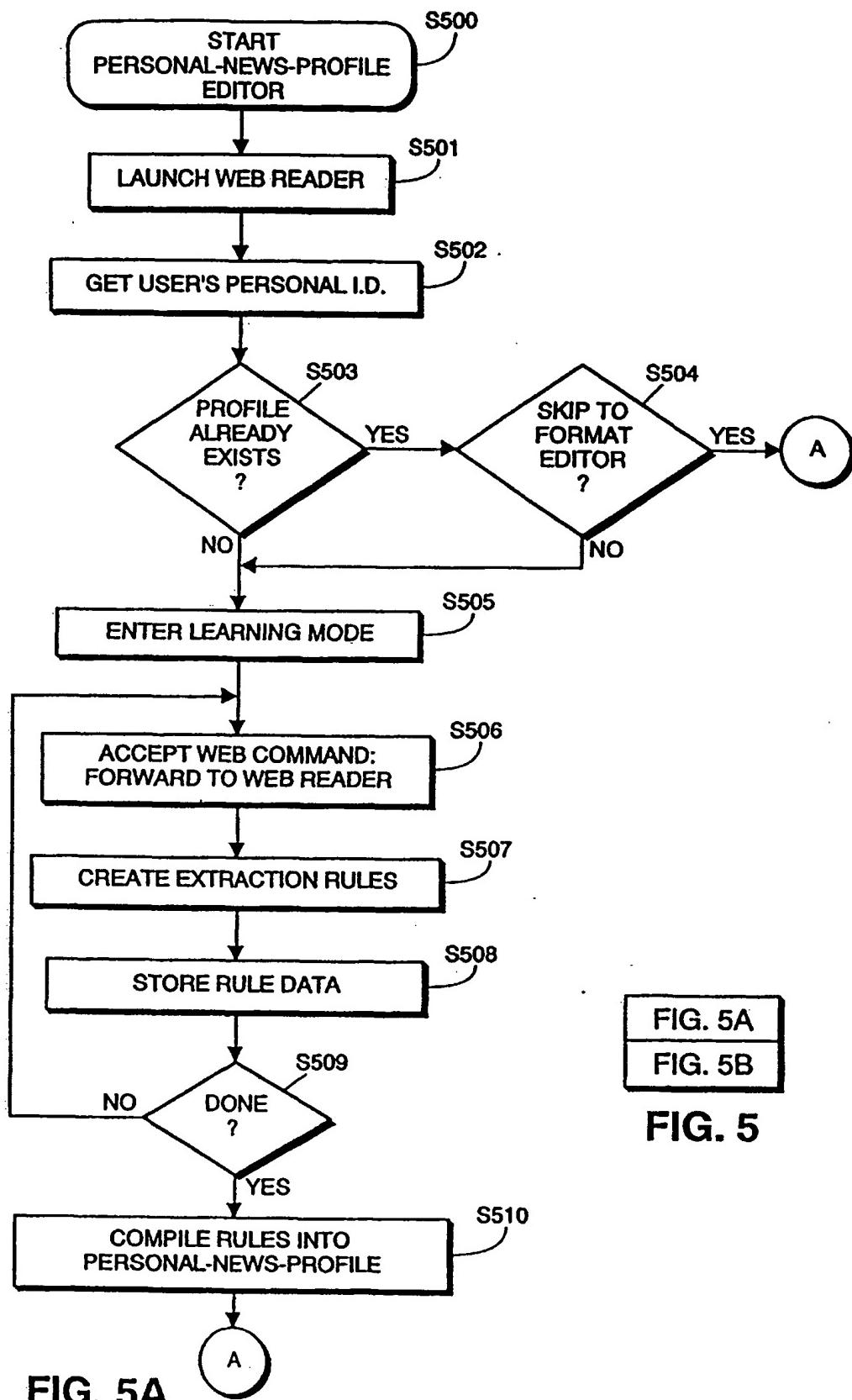


FIG. 4



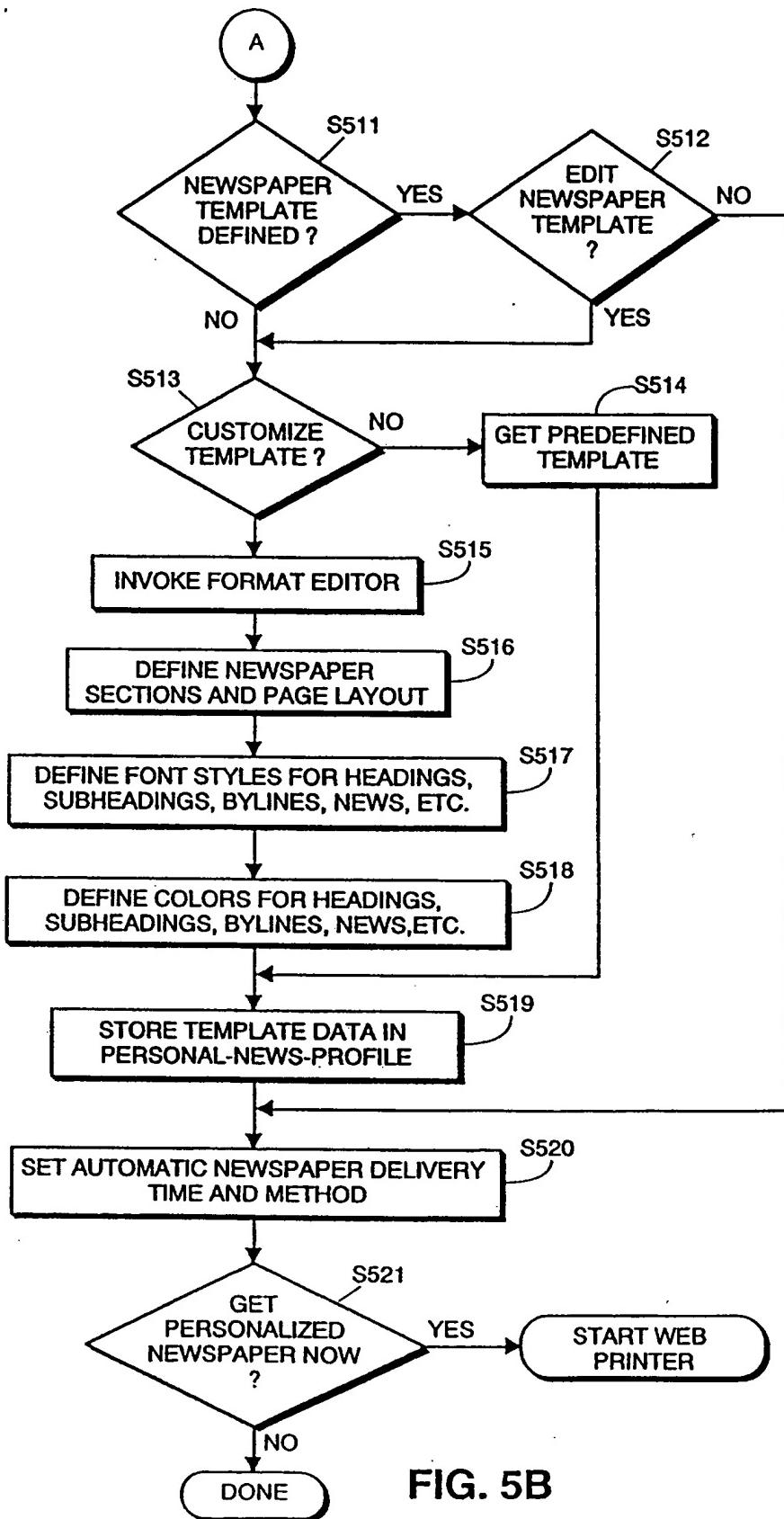


FIG. 5B

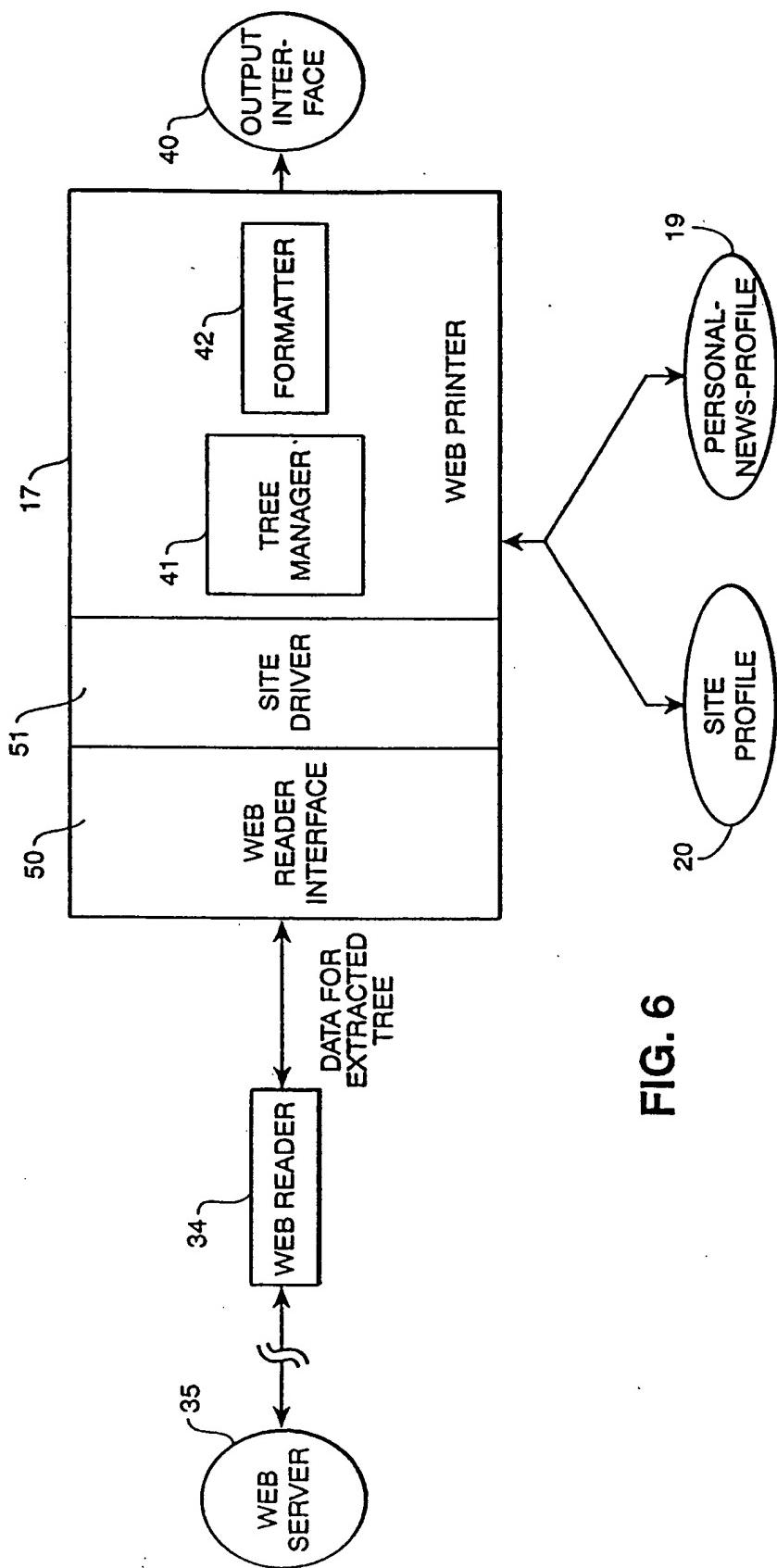


FIG. 6

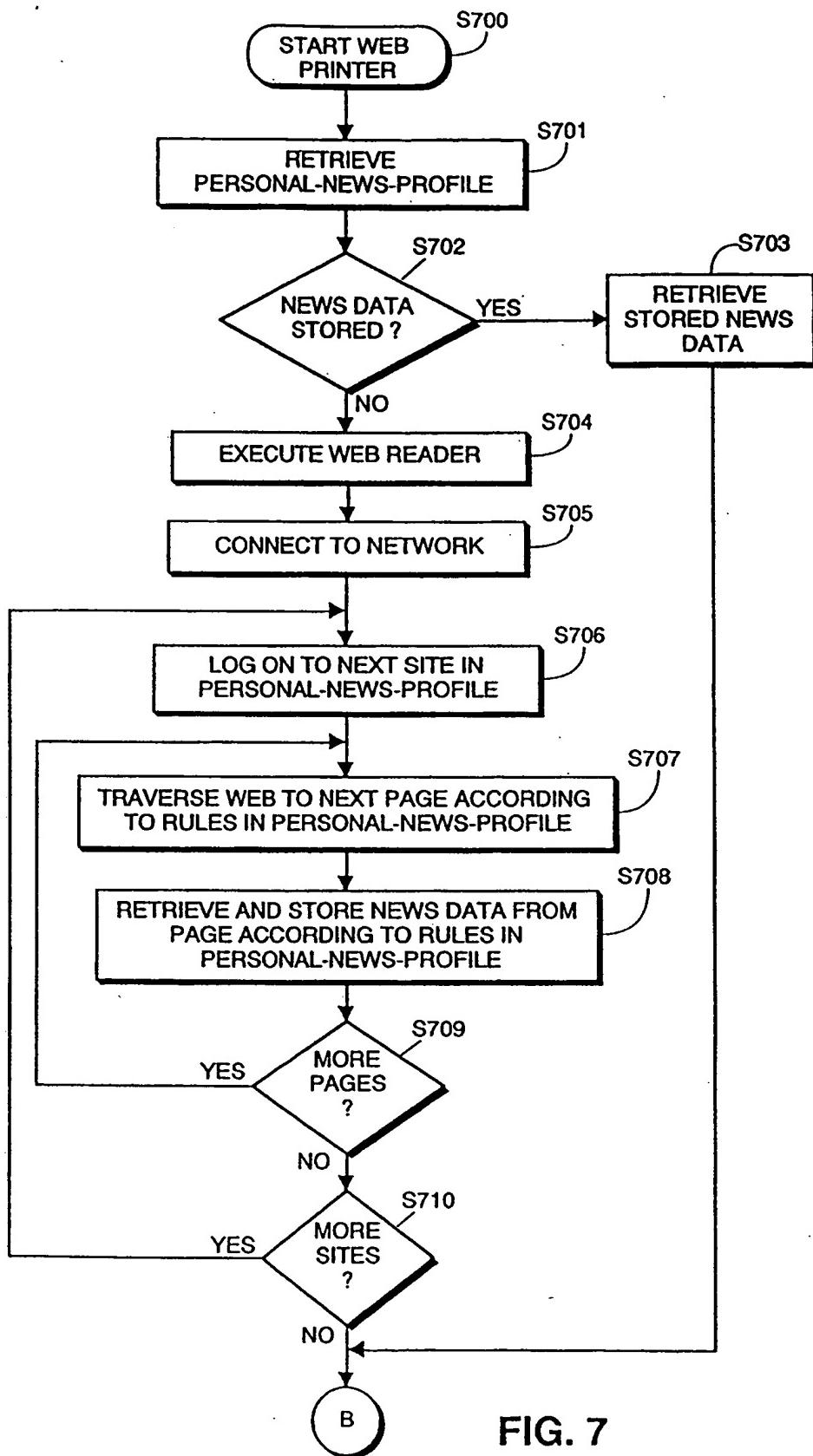


FIG. 7

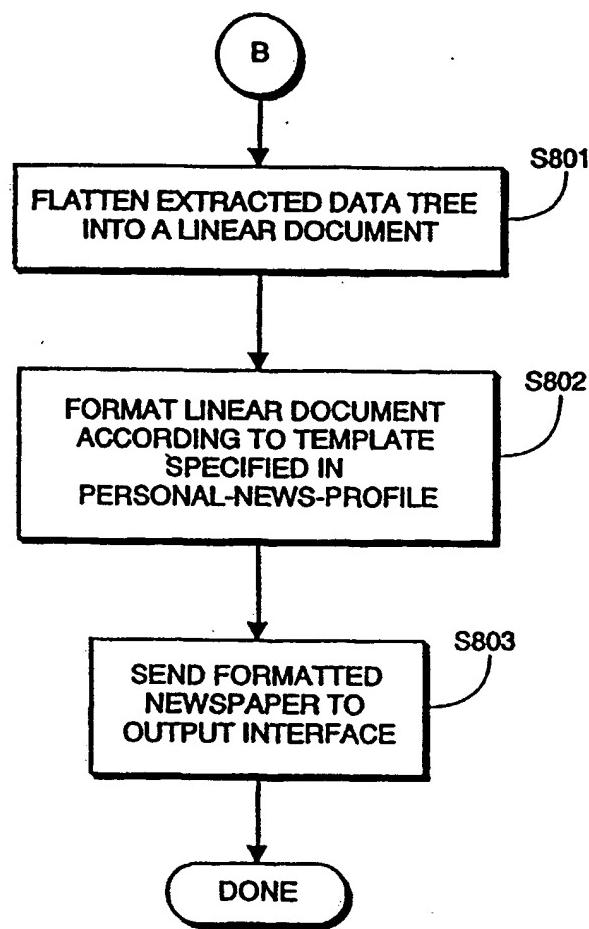


FIG. 8

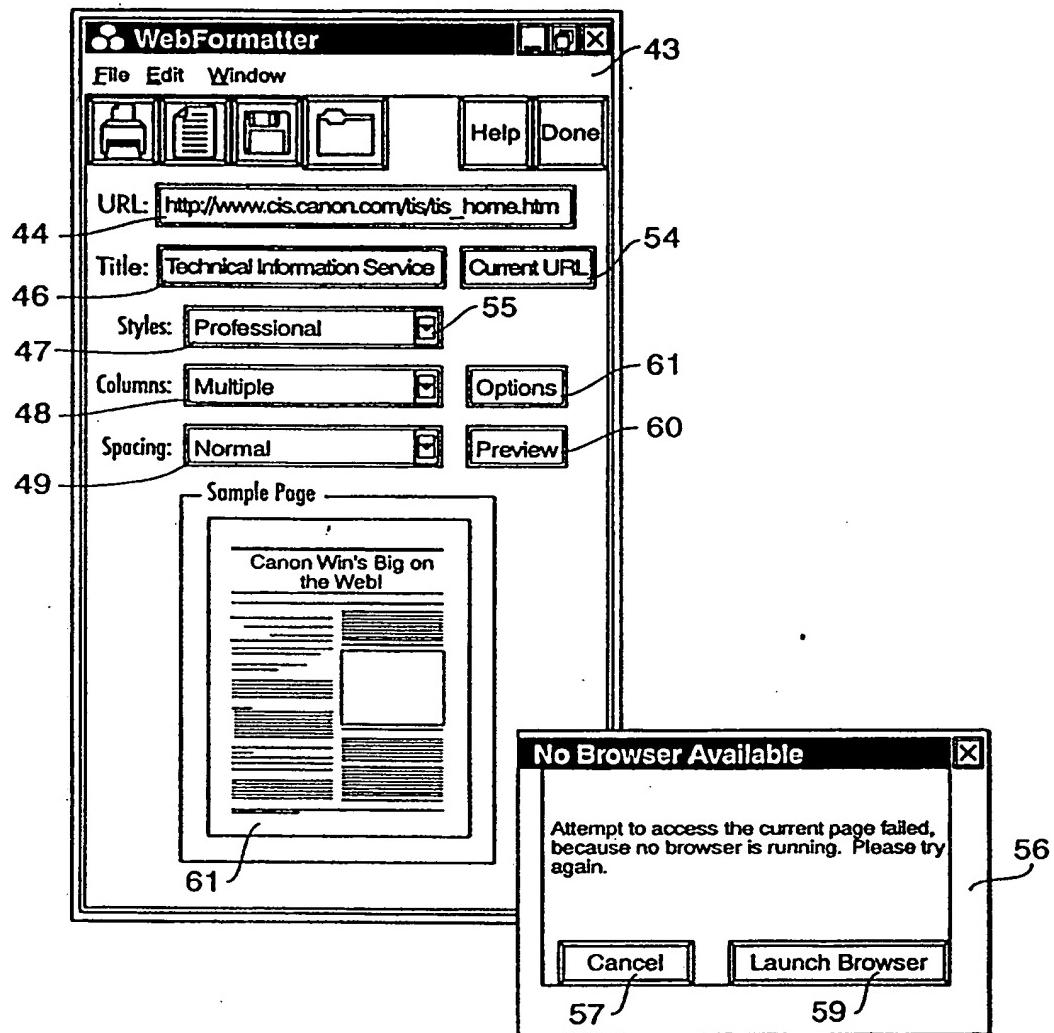


FIG.9A

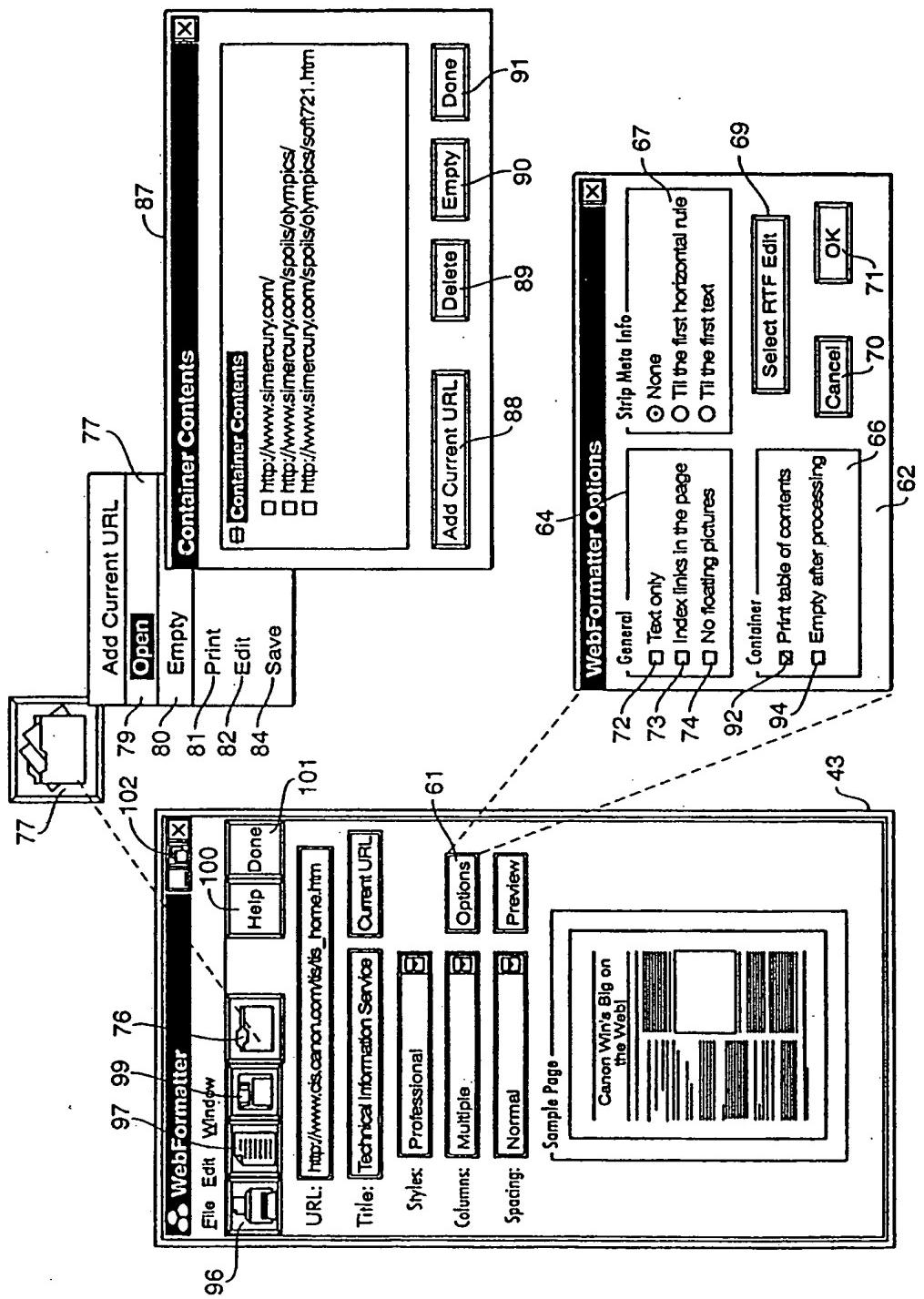


FIG.9B

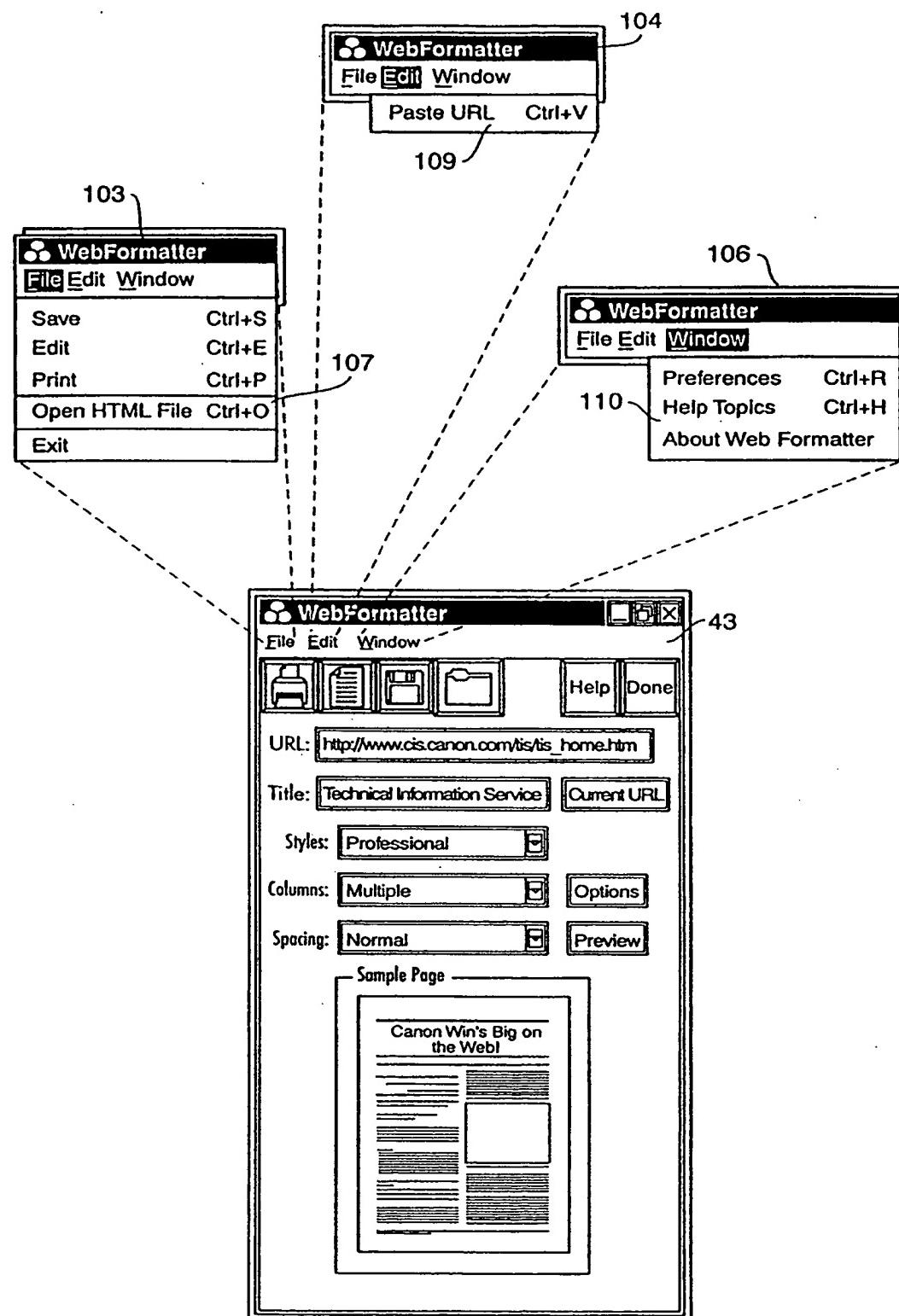


FIG.9C

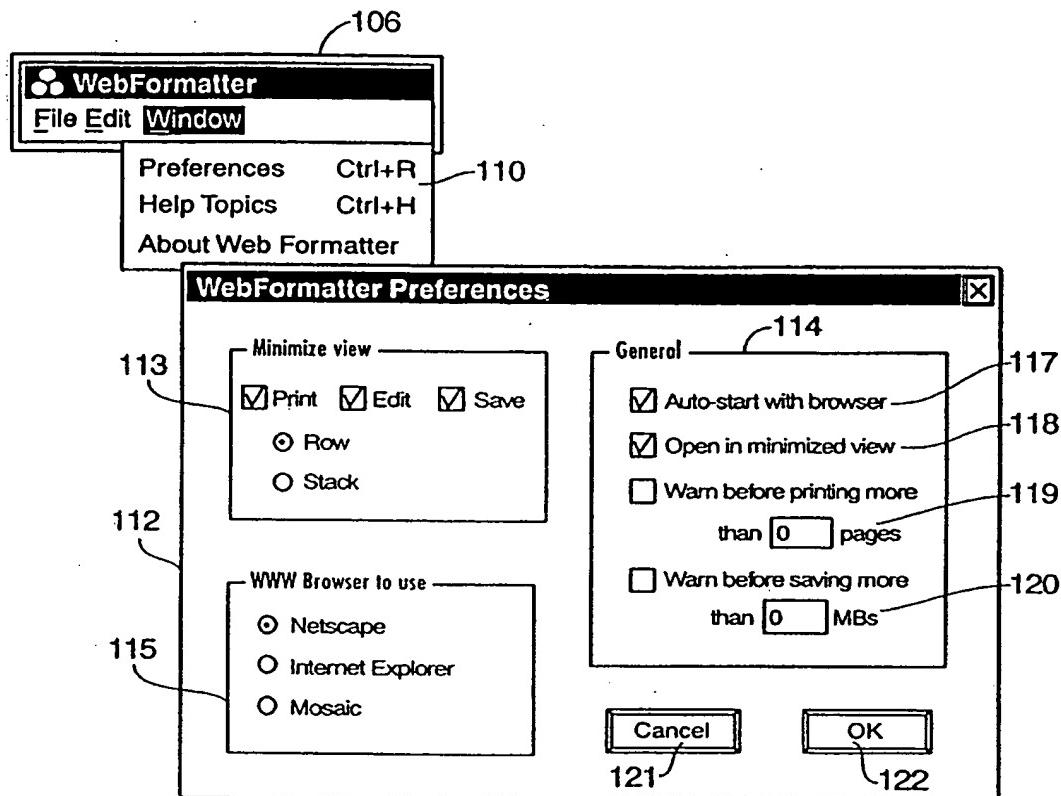


FIG.9D

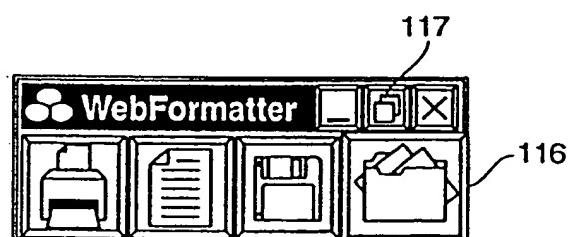


FIG.9E

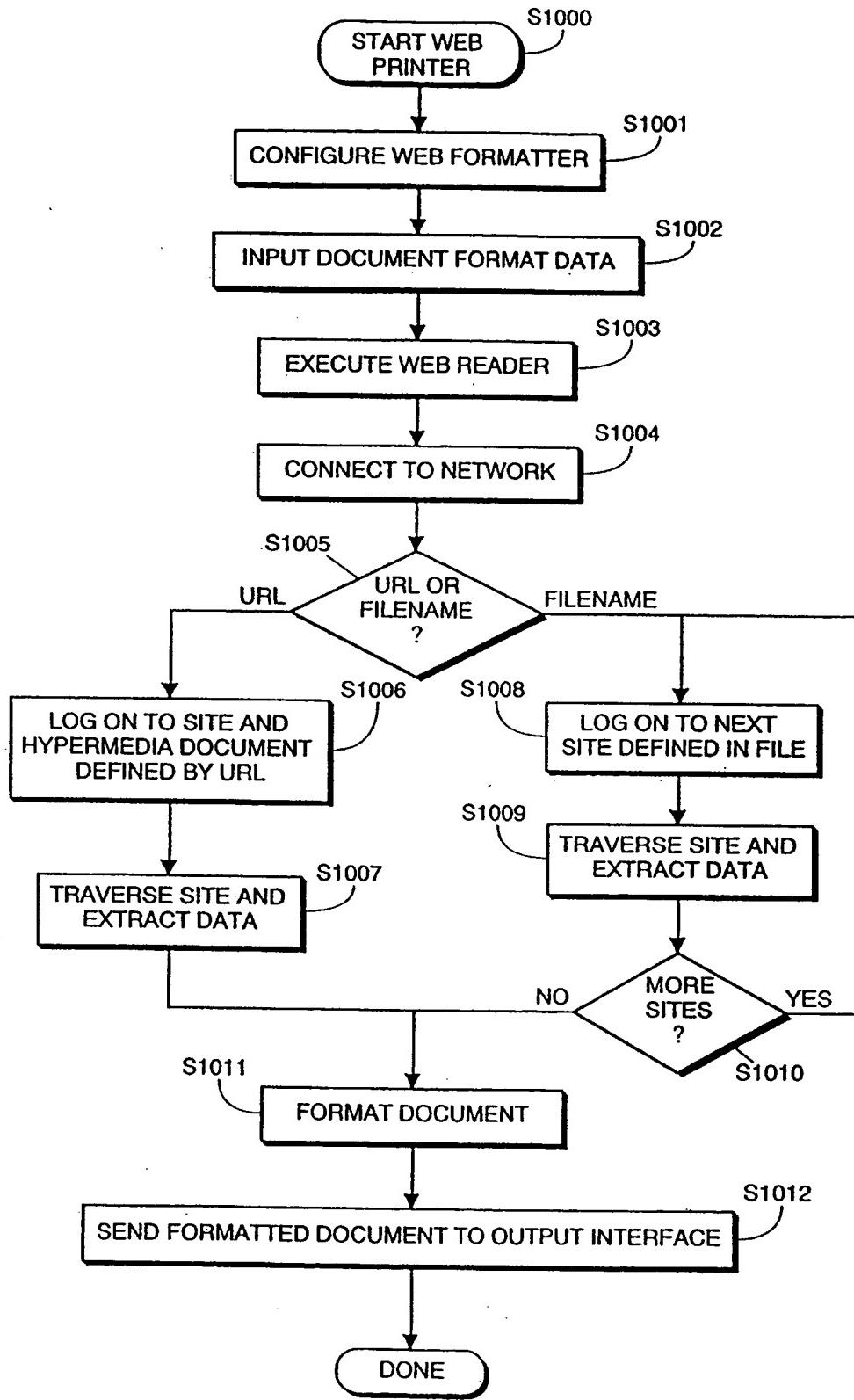


FIG. 10



(19)

Europäisches Patentamt

European Patent Office

Office européen des brevets



(11) EP 0 834 822 A3

(12)

EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
28.04.1999 Bulletin 1999/17(51) Int. Cl.⁶: G06F 17/30(43) Date of publication A2:
08.04.1998 Bulletin 1998/15

(21) Application number: 97307009.7

(22) Date of filing: 10.09.1997

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC
NL PT SEDesignated Extension States:
AL LT LV RO SI

(30) Priority: 04.10.1996 US 726853

(71) Applicant:
Canon Information Systems, Inc.
Costa Mesa, CA 92626 (US)(72) Inventors:
• Nehab, Smadar
Costa Mesa, California 92626 (US)
• Wickramaratne, Manjula G.
Costa Mesa, California 92626 (US)
• Klark, Paul L.
Costa Mesa, California 92626 (US)(74) Representative:
Beresford, Keith Denis Lewis et al
BERESFORD & Co.
2-5 Warwick Court
High Holborn
London WC1R 5DJ (GB)

(54) World wide web news retrieval system

(57) A World Wide Web site data retrieval system includes an input device for inputting data and commands to access the World Wide Web, and a memory for storing a Web site data retrieval driver which includes a Web reader, stored Web site address information, stored Web site commands, and stored format information. The memory also stores process steps to connect to a Web site and to issue commands within the connected Web site, and a connection to the World Wide Web. The system includes a processor for launching the Web site data retrieval driver in response to a command to access the World Wide Web. The Web site retrieval driver, upon being launched, (1) launches the

Web reader to connect to the World Wide Web via the connection, (2) retrieves the Web site address information and Web site commands, (3) instructs the Web reader to access the Web site based on the Web site address information and Web site commands, (4) downloads Web site data from the Web site based on the Web site commands, (5) stores the Web site data in a linear document, (6) repeats steps 1 through 5 until all addresses in the stored Web site address information have been accessed, and (7) formats the linear document into a personalized document based on the format information.

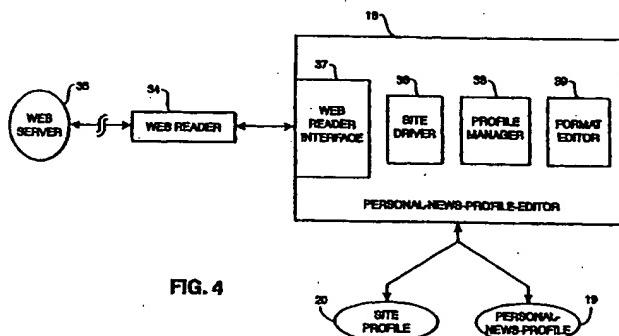


FIG. 4



European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 97 30 7009

DOCUMENTS CONSIDERED TO BE RELEVANT			CLASSIFICATION OF THE APPLICATION (Int.Cl.)
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	
X	EP 0 316 957 A (SHARP KK) 24 May 1989 * page 3, left-hand column, line 1 - right-hand column, line 35 *	1,2	G06F17/30
A	---	36-54	
A	KAMBA T ET AL: "An interactive, personalized, newspaper on the WWW" MULTIMEDIA COMPUTING AND NETWORKING 1996, SAN JOSE, CA, USA, 29-31 JAN. 1996, vol. 2667, pages 290-301, XP002094970 ISSN 0277-786X, Proceedings of the SPIE - The International Society for Optical Engineering, 1996, SPIE-Int. Soc. Opt. Eng, USA * page 292, paragraph 3.1 - page 298, paragraph 4.1 *	1-54	
A	CHESNAIS P R ET AL: "The Fishwrap personalized news system" PROCEEDINGS OF THE 2ND INTERNATIONAL WORKSHOP ON COMMUNITY NETWORKING INTEGRATED MULTIMEDIA SERVICES IN THE HOME (CAT. NO.TH8097), PROCEEDINGS OF THE SECOND INTERNATIONAL WORKSHOP ON COMMUNITY NETWORKING 'INTEGRATED MULTIMEDIA SERVICES TO THE HOME', , pages 275-282, XP002074902 ISBN 0-7803-2756-X, 1995, New York, NY, USA, IEEE, USA * page 275, right-hand column, paragraph II - page 280, right-hand column, paragraph VII; figures *	1-54	TECHNICAL FIELDS SEARCHED (Int.Cl.)
	---	-/--	G06F
The present search report has been drawn up for all claims			
Place of search	Date of completion of the search	Examiner	
THE HAGUE	26 February 1999	Fournier, C	
CATEGORY OF CITED DOCUMENTS		T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons A : technological background O : non-written disclosure P : intermediate document & : member of the same patent family, corresponding document	
X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document			



European Patent
Office

EUROPEAN SEARCH REPORT

Application Number
EP 97 30 7009

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.CI.6)
X	<p>LIEBERMAN H: "Letizia: an agent that assists Web browsing" IJCAI-95. PROCEEDINGS OF THE FOURTEENTH INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, PROCEEDINGS OF INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, MONTREAL, QUE., CANADA, 20-25 AUG. 1995, pages 924-929 vol.1, XP002094971 1995, San Mateo, CA, USA, Morgan Kaufmann Publishers, USA; http://ieber-www.media.mit.edu/people/ieber/Lieberary/Letizia/Letizia.html * page 925, left-hand column, paragraph 3 - page 927, left-hand column, paragraph 6 *</p> <p>-----</p>	7,10	
TECHNICAL FIELDS SEARCHED (Int.Cl.6)			
<p>The present search report has been drawn up for all claims</p>			
Place of search	Date of completion of the search	Examiner	
THE HAGUE	26 February 1999	Fournier, C	
CATEGORY OF CITED DOCUMENTS		T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons A : member of the same patent family, corresponding document	
X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document			

**ANNEX TO THE EUROPEAN SEARCH REPORT
ON EUROPEAN PATENT APPLICATION NO.**

EP 97 30 7009

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report.
The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

26-02-1999

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0316957 A	24-05-1989	JP 1134561 A	26-05-1989

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.